PRÁCTICA 2: Limpieza y validación de los datos

Carles Colomina

23 de diciembre de 2019

Contents

1 Selección del dataset de trabajo. 1

2 Descripción del dataset. 2
2.1 Autores y descripción del dataset. . . . 2
2.2 Importancia del dataset y preguntas que se pretende responder en esta práctica. . . . 3
2.3 Análisis descriptivo inicial. . . . 3

3 Limpieza del dataset. 18
3.1 Integración de los datos. . . . 18
3.2 Selección de datos de interés. . . . 18
3.3 Reducción. . . . 19
3.4 Conversión. . . . 19
3.5 Ceros y datos vacíos. . . . 20
3.6 Tratamiento de valores extremos. . . . 21

4 Análisis de los datos. 22
4.1 Análisis descriptivo. . . . 22
4.2 Selección de los grupos de datos que se quieren comparar. . . . 28
4.3 Comprobación de normalidad y homocedasticidad de las variables. . . . 29
4.4 Pruebas estadísticas de comparación de variables. . . . 41
4.5 Correlación entre variables. . . . 49
4.6 Modelos de regresión lineal. . . . 52
4.7 Modelos de regresión logística propuestos. . . . 53
4.8 Modelos supervisados propuestos (Random Forest). . . . 55

5 Representación de los resultados. 57
5.1 Boxplots de los atributos de los vinos blancos y tintos. . . . 57
5.2 Correlaciones. . . . 59
5.3 Tabla resumen tests estadísticos. . . . 61
5.4 Proyecciones PCA: . . . 62
5.5 Tablas resumen modelos de análisis. . . . 64
5.6 Conclusiones, resolución a las preguntas planteadas. . . . 65

1 Selección del dataset de trabajo.

El dataset de trabajo seleccionado es el "Red & White Wine Dataset", ya que es interesante para testear modelos de regresión y clasificación. Otro motivo para seleccionar este dataset, es que en otras asignaturas del máster ya he trabajado con otros datasets de Kaggle, por ejemplo con el "Titanic", por contra nunca he trabajado con el "Wine Quality Data Set", por lo que es una buena oportunidad para aprender y reflexionar sobre los resultados que se obtengan.

Los datos para realizar la práctica se han descargado del repositorio Kaggle: Red White Wine Dataset. En la práctica se ha propuesto trabajar a modo de ejemplo con el "Red Dataset", en mi caso prefiero trabajar con ambos datasets, el de vino blanco y el de vino tinto, ya que esto hace más interesante la práctica ya que proporciona mayores posibilidades de análisis.

Procedemos a seleccionar el directorio de trabajo y a cargar los datos:

```
setwd("C:/Users/Carlos/Desktop/Ciencia de Datos/Tipología y ciclo de vida de los datos/PRACTICA 2")

data <- read.csv("wine_dataset.csv", header = TRUE, sep = ",")

#Separamos los datasets en tintos y blancos por si más adelante queremos
#realizar algún tipo de análisis específico para cada tipo de vino.

data_red <- data[which(data$style == "red"), ]
data_white <- data[data$style == "white", ]
```

2 Descripción del dataset.

2.1 Autores y descripción del dataset.

Realizamos primero una descripción del dataset, y de las variables que lo componen.

La descripción del dataset es importante, ya que nos permite entender los datos de que disponenmos, sus características, su relevancia en el dataset y el motivo por el que los autores los han seleccionado; se nos indican también los métodos con los que han sido recopilados (esto nos permite entender por ejemplo las posibles causas de valores nulos o vacíos, etc.) y los objetivos que se plantean los autores de los datos en el momento de recopilarlos.

Una buena comprensión inicial del dataset es fundamental para extraer información relevante del mismo.

El dataset "Red & White Wine Dataset", presenta dos datasets, uno de vino blanco y otro de vino tinto (identificados como 0blanco y 1: tinto), con una serie de datos químico físicos de los mismos. Todas las variables numéricas que incluye son bastante relevantes según los autores en la calidad final de un vino. Finalmente el dataset dispone de una variable cualitativa categórica que nos indica como es clasificado cada vino por expertos entre 0-10 (muy mala calidad - muy buena calidad).

Citación de los autores.

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties.

- [Elsevier] (http://dx.doi.org/10.1016/j.dss.2009.05.016)

Las **variables** del dataset son:

1 - fixed acidity: los ácidos fijados son aquellos que no son volátiles y permanecen en el vino, se miden en mg/l.

2 - volatile acidity: los ácidos volátiles tienen mayor volatilidad e influyen en el aroma del vino, se miden en mg/l.

3 - citric acid: concentración de ácido cítrico en mr/l.

4 - residual sugar: concentración de azúcares en mg/l.

5 - chlorides: concentración de cloruros en mg/l.

6 - free sulfur dioxide: concentración libre de dióxido de azufre en mg/l.

7 - total sulfur dioxide: concentración total de dióxido de azufre en mg/l.

8 - density: densidad del vino en gr/cm3.

9 - pH: pH del vino.

10 - sulphates: concentración de sulfatos del vino en mg/l.

11 - alcohol: graduación alcohólica del vino en grados.

12 - quality (score between 0 and 10): escala cualitativa de calidad determinada por expertos, 0 indica calidad pésima, 10 calidad excelente.

13 - style: indica el tipo de vino, 1- red, 0- white.

2.2 Importancia del dataset y preguntas que se pretende responder en esta práctica.

La importancia o utilidad del dataset deriva, en que nos aporta un panel datos que nos permitirán evaluar si es posible partiendo de datos físico químicos previos, poder determinar la calidad final de un vino de forma similar a como lo haría un panel de expertos. Además, de los métodos de análisis evaluados, nos permitirá determinar cuales son los mejores para llevar a cabo estas predicciones.Una vez se disponga de métodos de análisis con adecuadas medidas de rendimiento,estos análisis podrán ser empleados por los fabricantes de vinos para poder mejorar la calidad de los mismos, enfocándose en los conjuntos de variables físico químicas que sean más relevantes en la cildad final del vino.

Las preguntas básicas de partida que nos plantearemos serán las siguientes:

- ¿Tiene sentido trabajar con todos los vinos en conjunto o es mejor trabajar con ellos de forma separada (blancos respecto a tintos)?,¿son significativamente distintos los vinos blancos de los tintos?
- ¿Existe algún parámetro químico físico que sea especialmente relevante para diferenciar vinos de buena calidad?
- ¿Es posible predecir, por ejemplo mediante regresión lineal, u otros tipos de análisis, en base a los datos químico físicos disponibles de un vino, determinar la calidad resultante del mismo antes de que lo valore un panel de expertos?, ¿Cual sería el mejor de estos métodos de análisis?

2.3 Análisis descriptivo inicial.

Realizaremos primero un análisis descriptivo previo antes de realizar las tareas propias de limpieza. Este análisis previo es interesante para ver como cambia el dataset una vez realizadas las tareas de limpieza.

```
str(data)
```

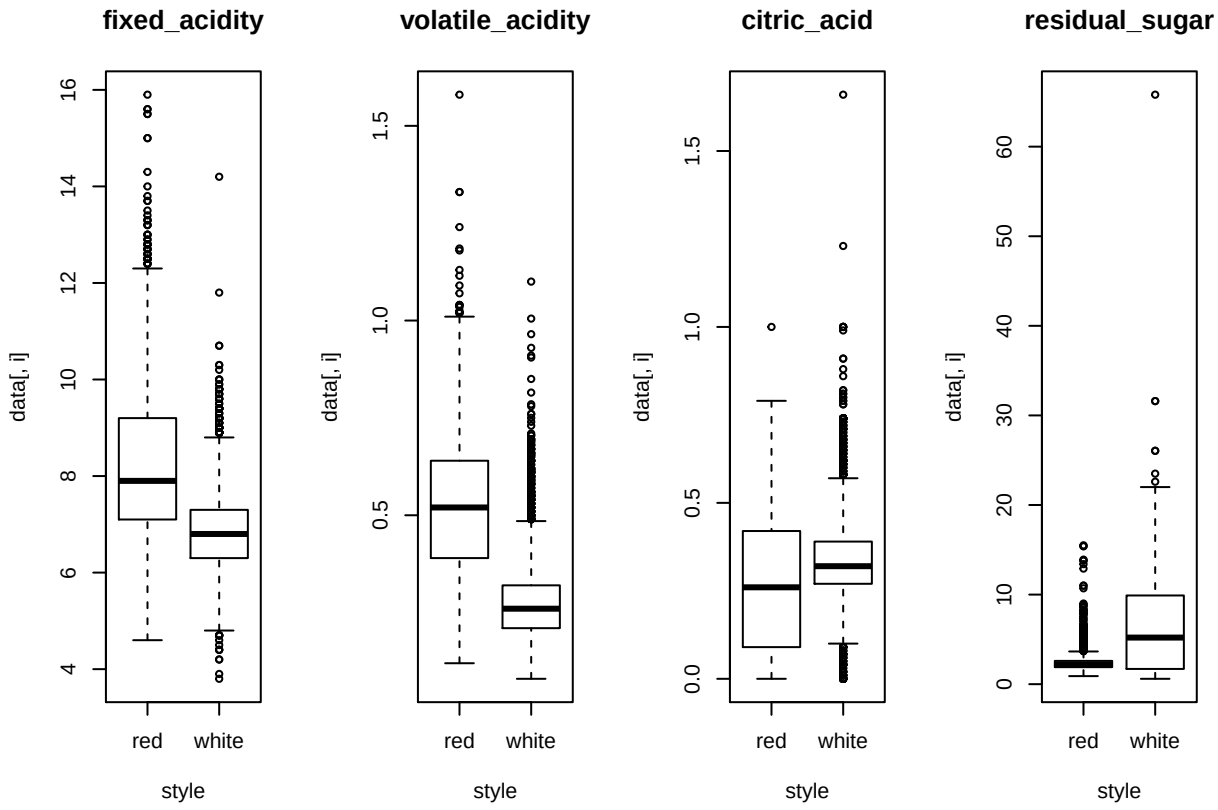
```
## 'data.frame':    6497 obs. of  13 variables:
##  $ fixed_acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile_acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric_acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual_sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free_sulfur_dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total_sulfur_dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
##  $ style               : Factor w/ 2 levels "red","white": 1 1 1 1 1 1 1 1 1 1 ...
```

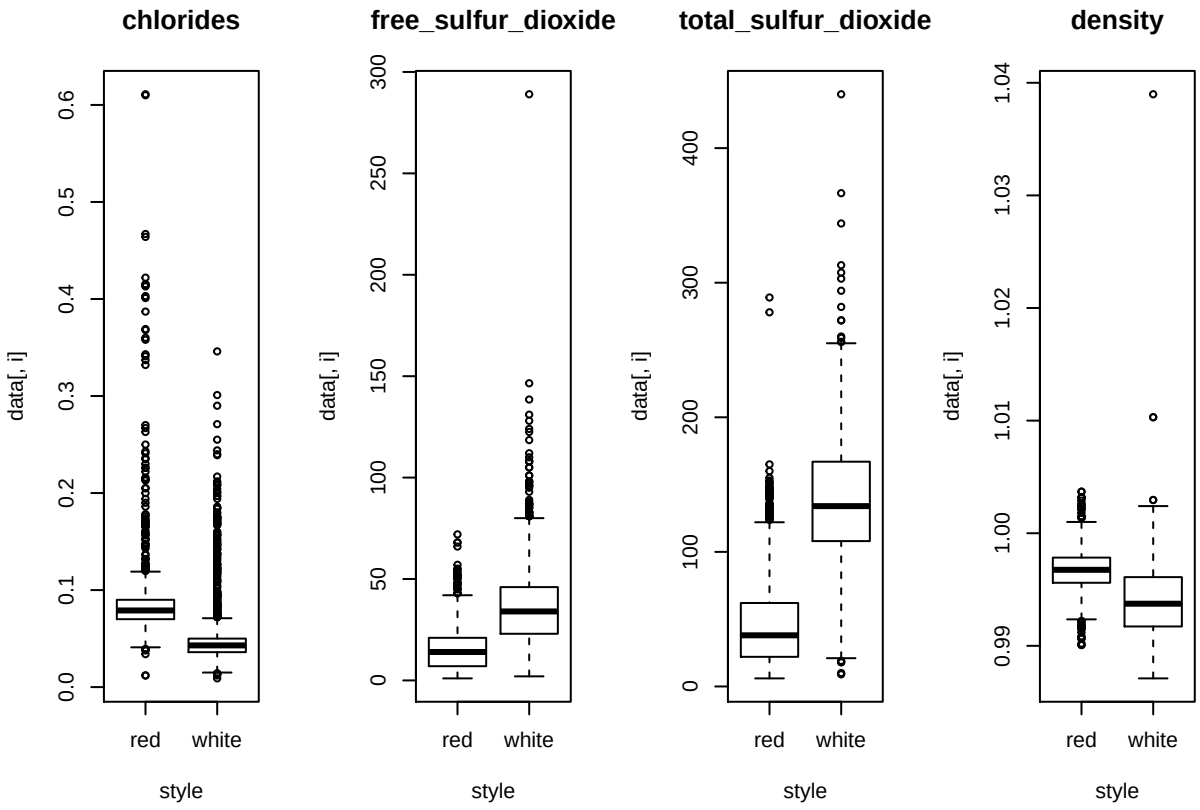
```
summary(data)
```

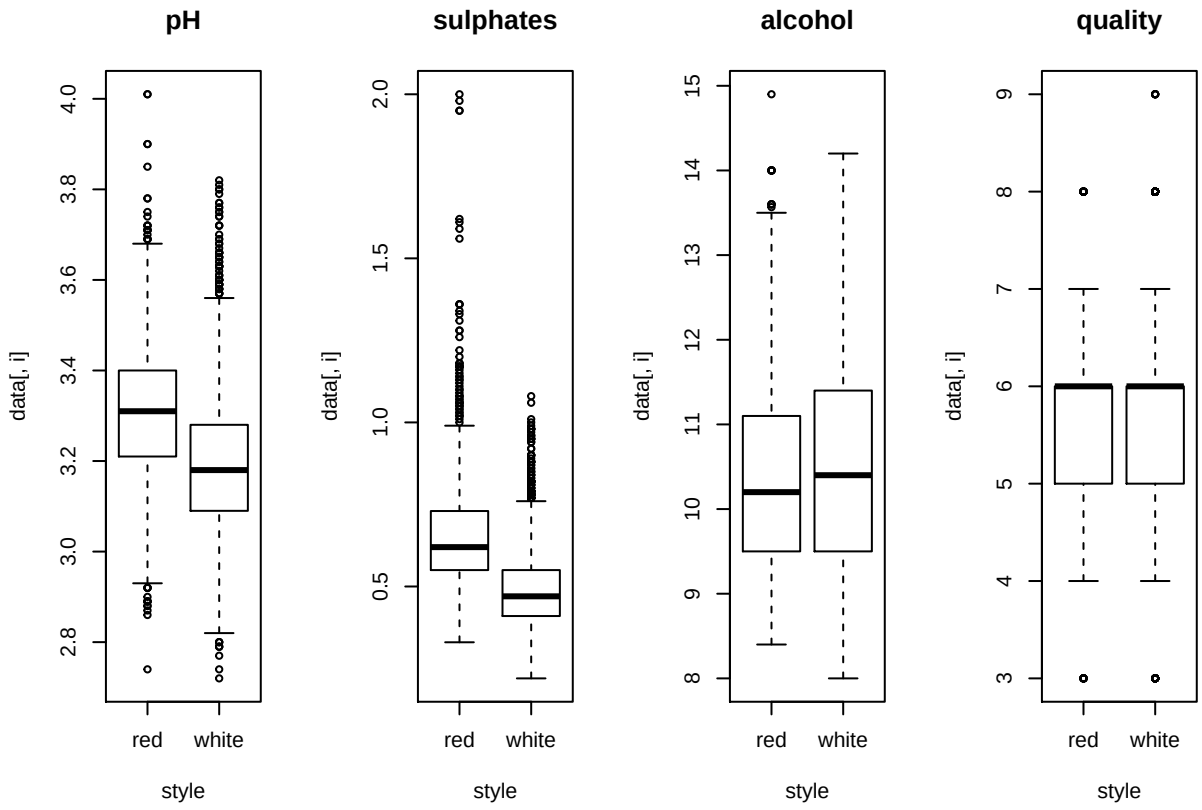
```
##  fixed_acidity    volatile_acidity  citric_acid     residual_sugar
```

```
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides       free_sulfur_dioxide total_sulfur_dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality        style
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000   red  :1599
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000   white:4898
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
```

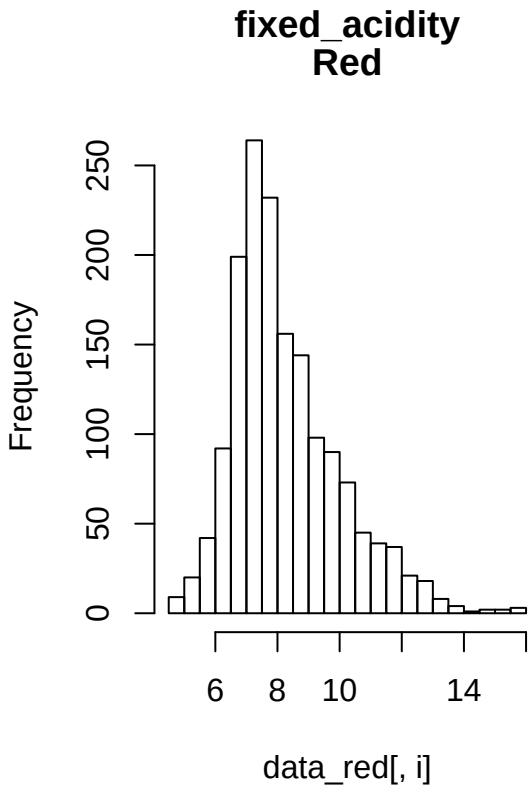
```
#Generamos boxplots de las variables separando los blancos de los tintos.
par(mfrow=c(1,4))
for (i in 1:(ncol(data)-1)) {
    boxplot(data[,i] ~ style, data=data, main=c(names(data[i])))
}
```

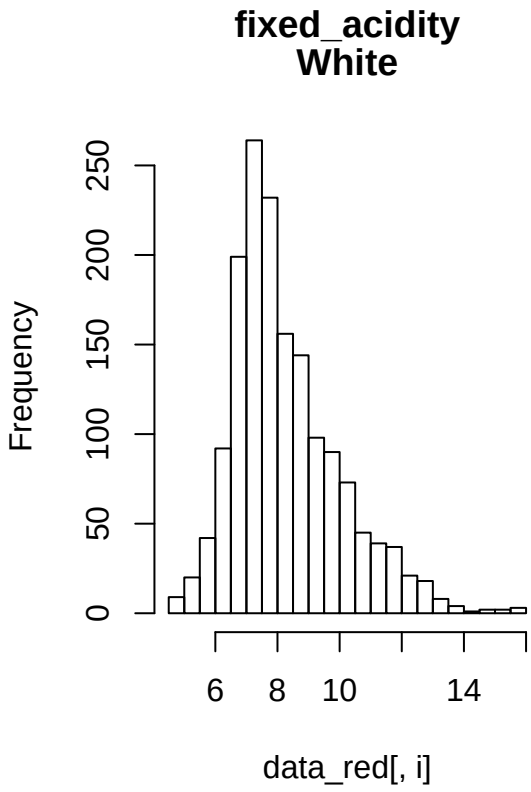


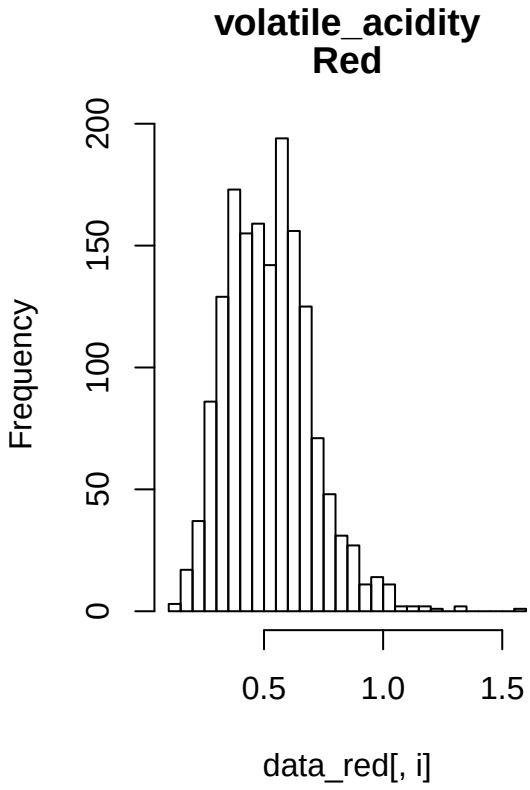

chlorides
data[, i]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
red
white
style
free_sulfur_dioxide
data[, i]
0
50
100
150
200
250
300
red
white
style
total_sulfur_dioxide
data[, i]
0
100
200
300
400
red
white
style
density
data[, i]
0.99
1.00
1.01
1.02
1.03
1.04
red
white
style

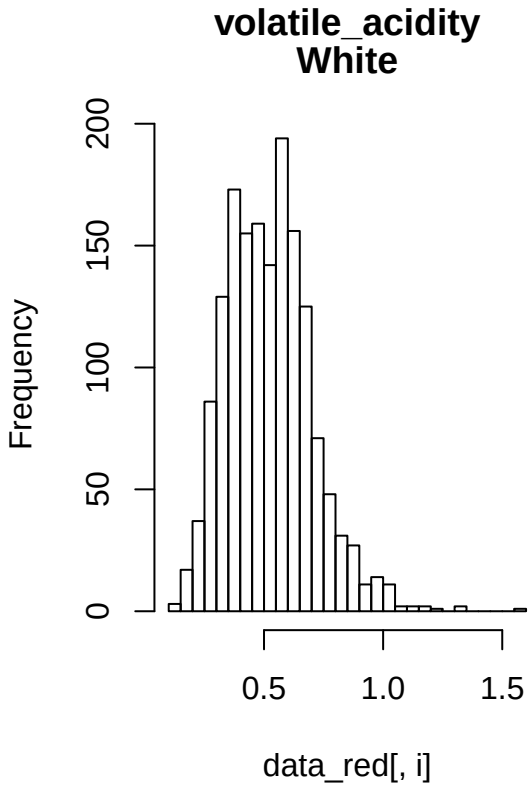


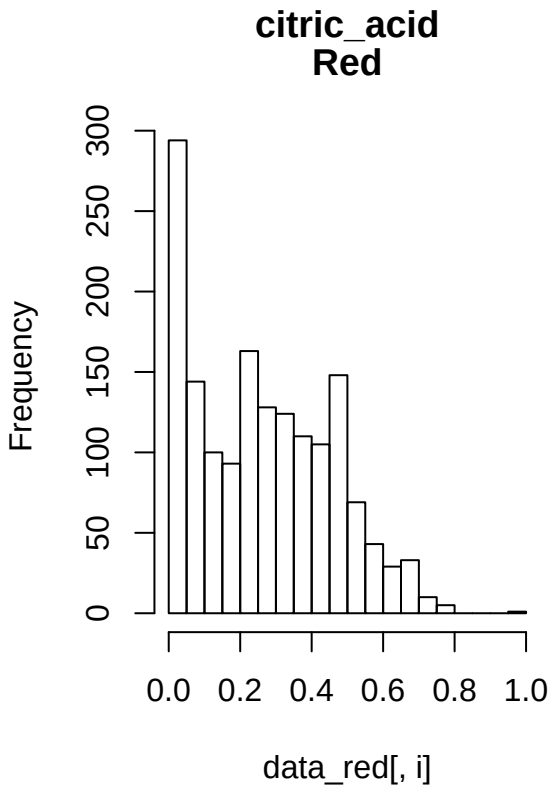

```
#Realizamos también histogramas de las diferentes variables.
par(mfrow=c(1,2))
for (i in 1:(ncol(data)-1)) {
        hist(data_red[,i], breaks=25, main=c(names(data[i]),"Red"))
        hist(data_red[,i], breaks=25, main=c(names(data[i]), "White"))
}
```

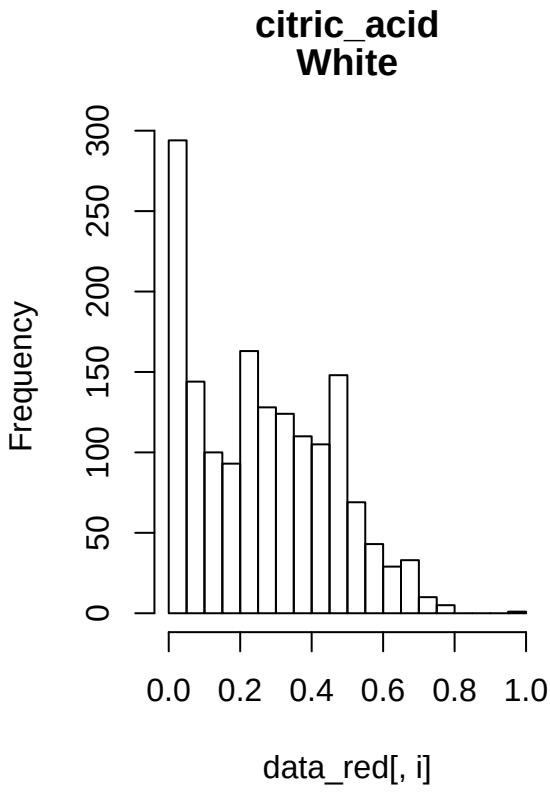

fixed_acidity
Red
Frequency
0
50
100
150
200
250
6
8
10
14
data_red[, i]

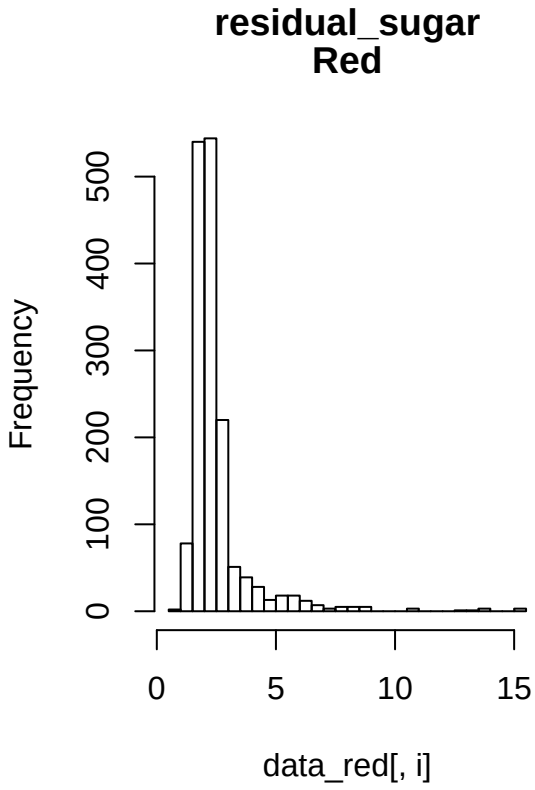

fixed_acidity
White
Frequency
0
50
100
150
200
250
6
8
10
14
data_red[, i]

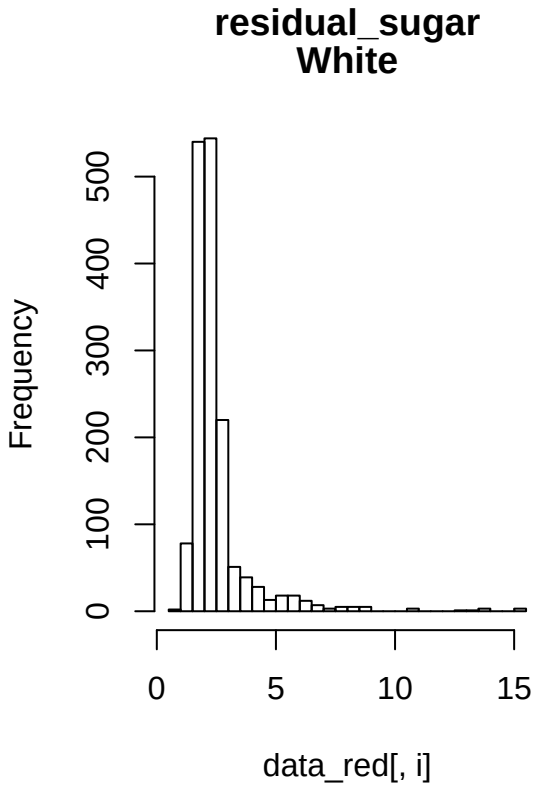

volatile_acidity
Red
Frequency
0
50
100
150
200
0.5
1.0
1.5
data_red[, i]

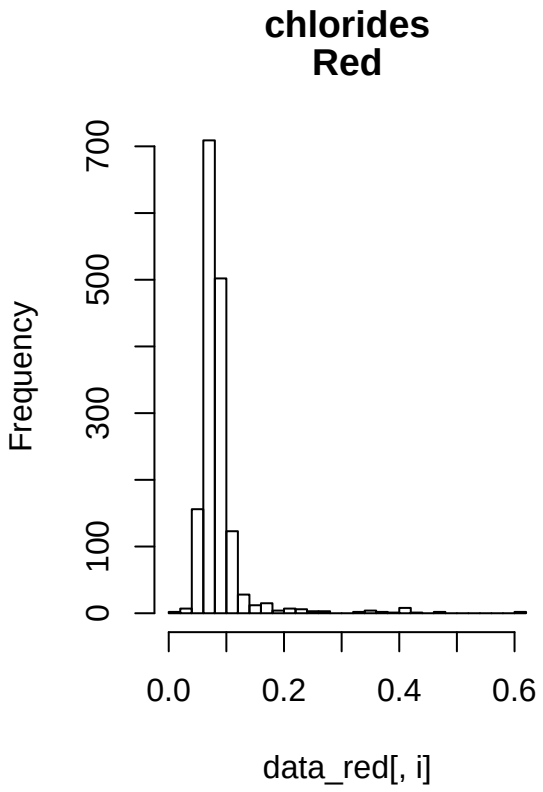

volatile_acidity
White
Frequency
0
50
100
150
200
0.5
1.0
1.5
data_red[, i]

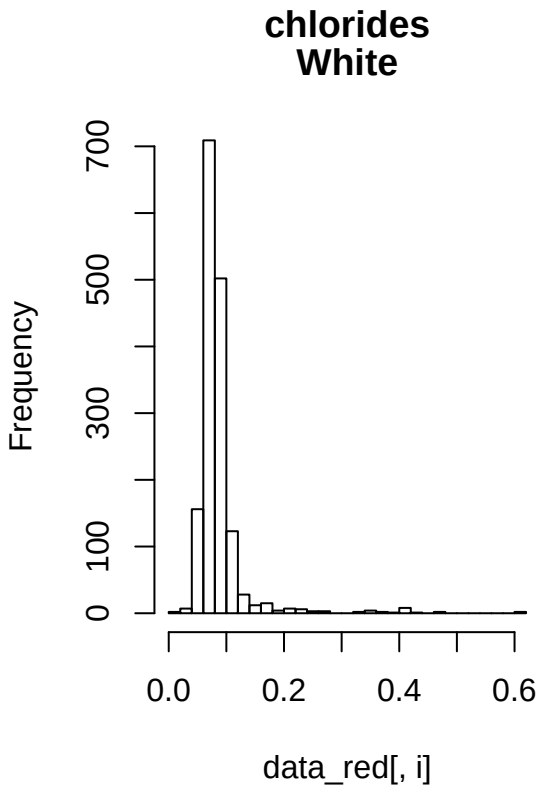

citric_acid
Red
Frequency
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0
data_red[, i]

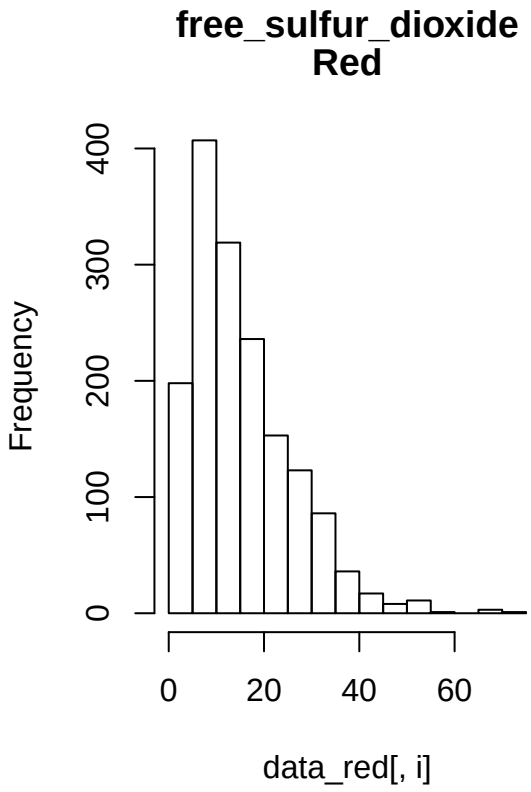

citric_acid
White
Frequency
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0
data_red[, i]

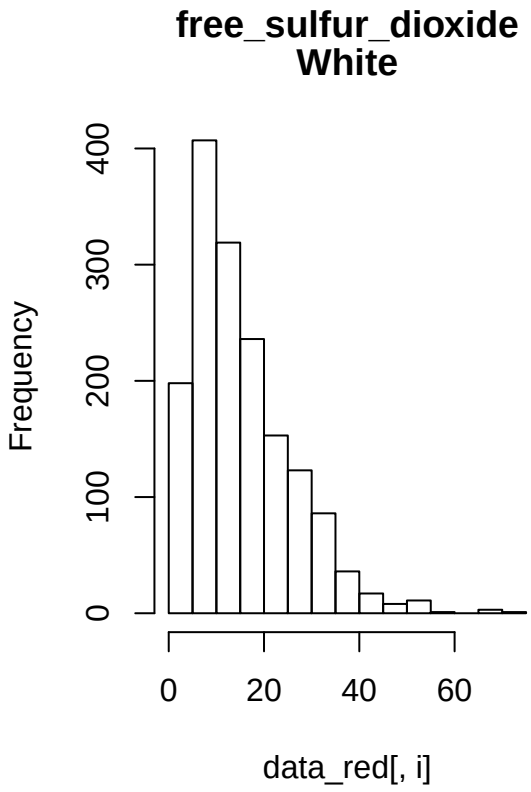

residual_sugar
Red
Frequency
0
100
200
300
400
500
0
5
10
15
data_red[, i]

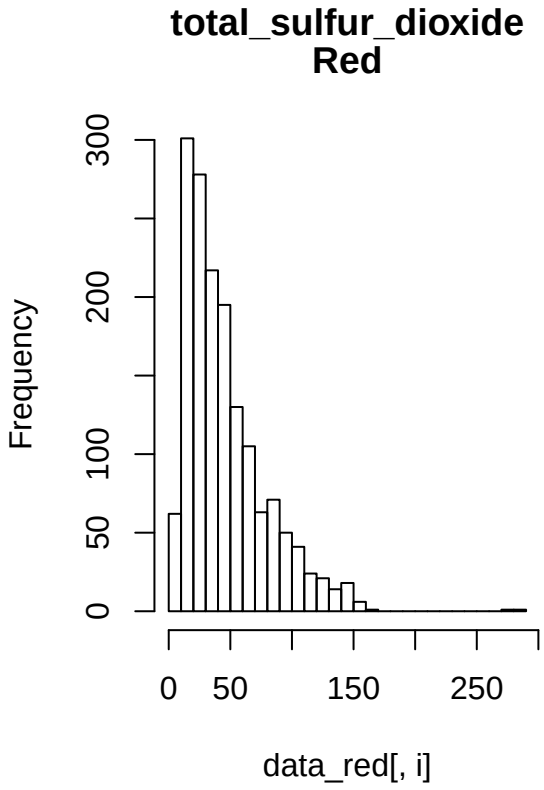

residual_sugar
White
Frequency
0
100
200
300
400
500
0
5
10
15
data_red[, i]

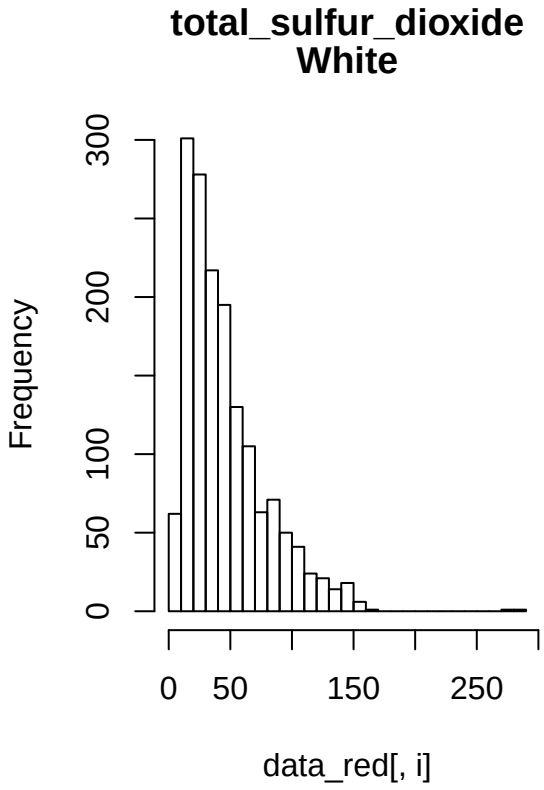

chlorides
Red
Frequency
0
100
300
500
700
0.0
0.2
0.4
0.6
data_red[, i]

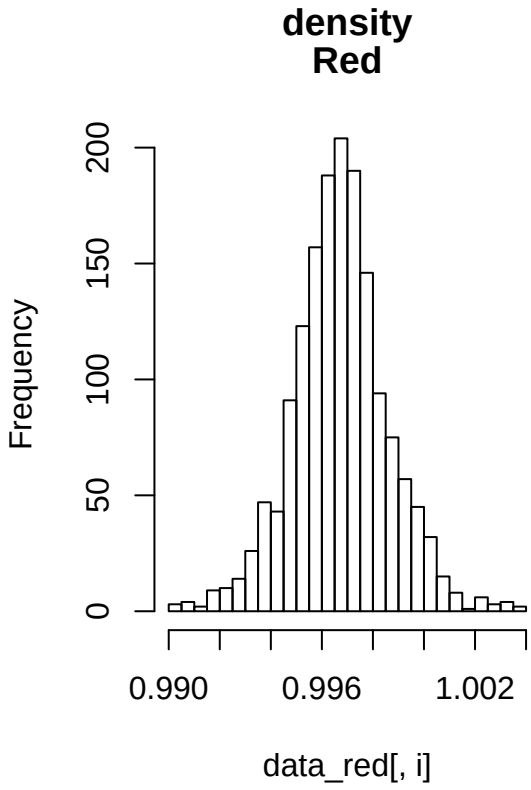

chlorides
White
Frequency
0
100
300
500
700
0.0
0.2
0.4
0.6
data_red[, i]

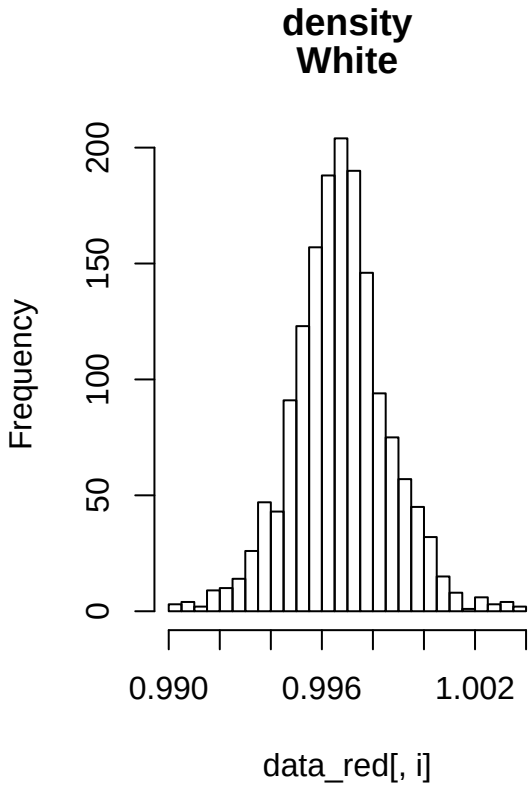

free_sulfur_dioxide
Red
Frequency
0
100
200
300
400
0
20
40
60
data_red[, i]

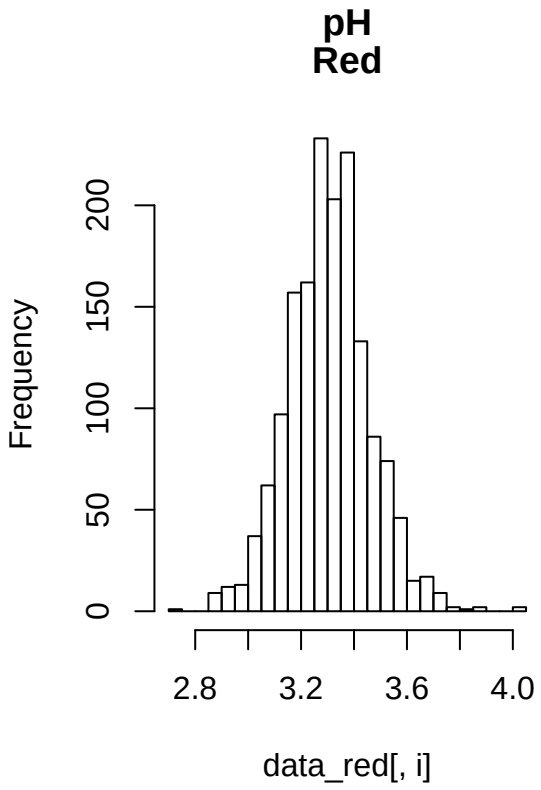

free_sulfur_dioxide
White
Frequency
0
100
200
300
400
0
20
40
60
data_red[, i]

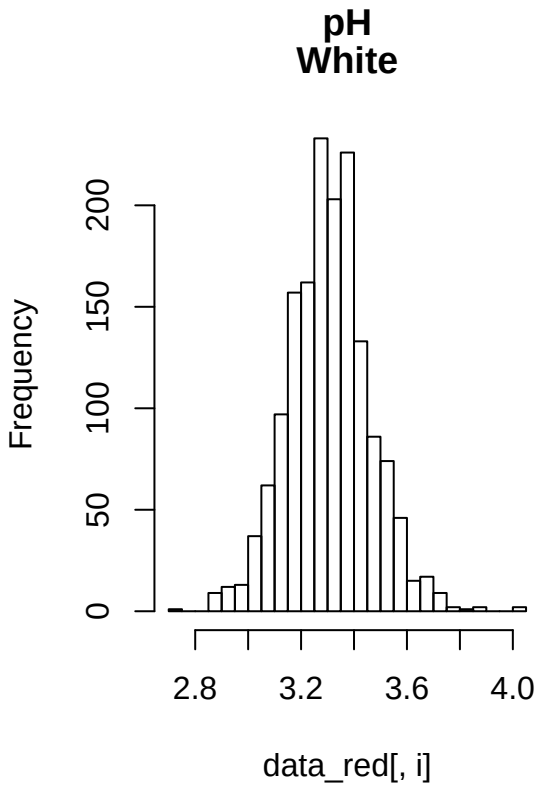

total_sulfur_dioxide
Red
Frequency
0
50
100
200
300
0
50
150
250
data_red[, i]

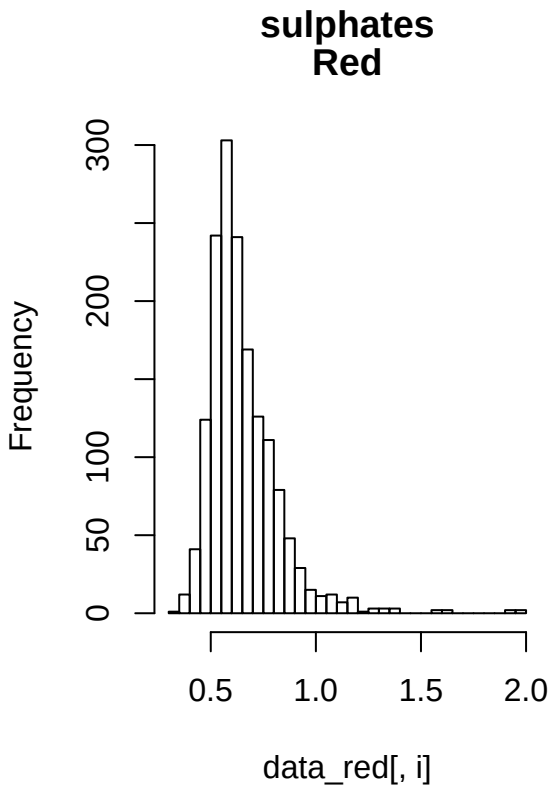

total_sulfur_dioxide
White
Frequency
0
50
100
200
300
0
50
150
250
data_red[, i]

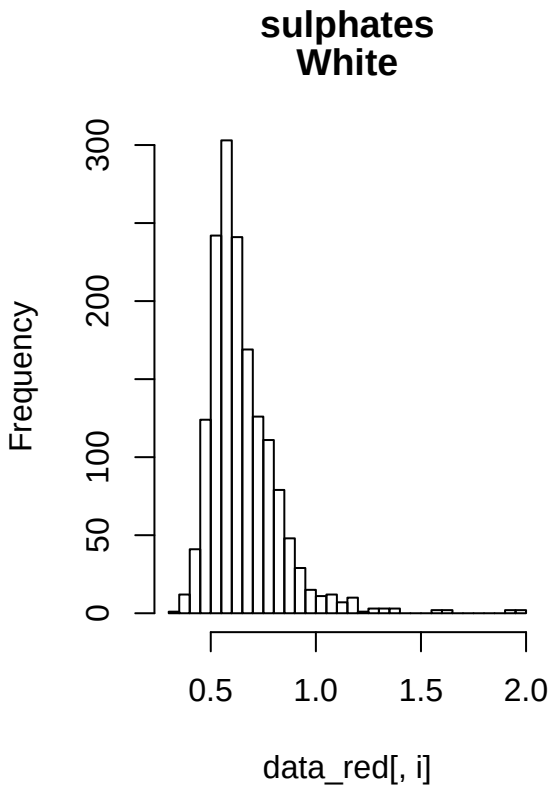

density
Red
Frequency
0
50
100
150
200
0.990
0.996
1.002
data_red[, i]

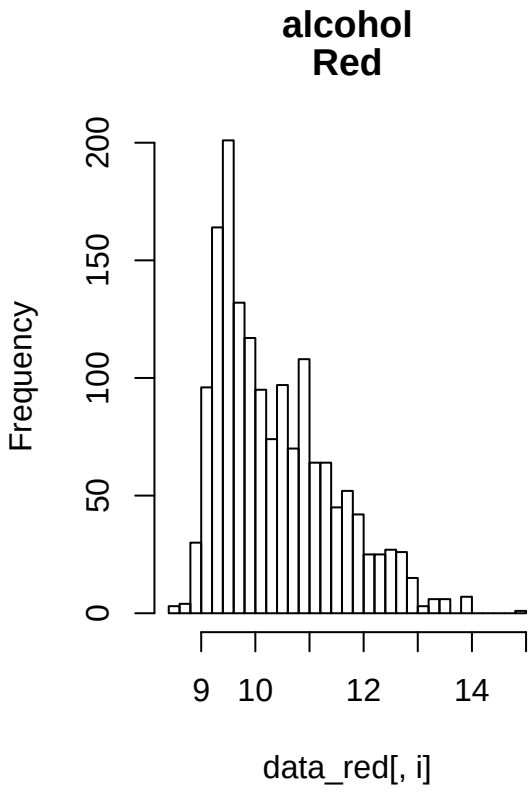

density
White
Frequency
0
50
100
150
200
0.990
0.996
1.002
data_red[, i]

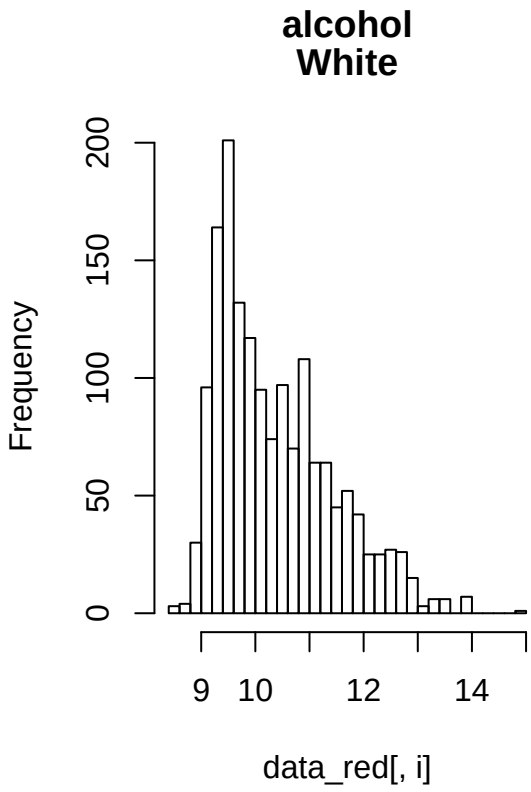

pH
Red
Frequency
0
50
100
150
200
2.8
3.2
3.6
4.0
data_red[, i]

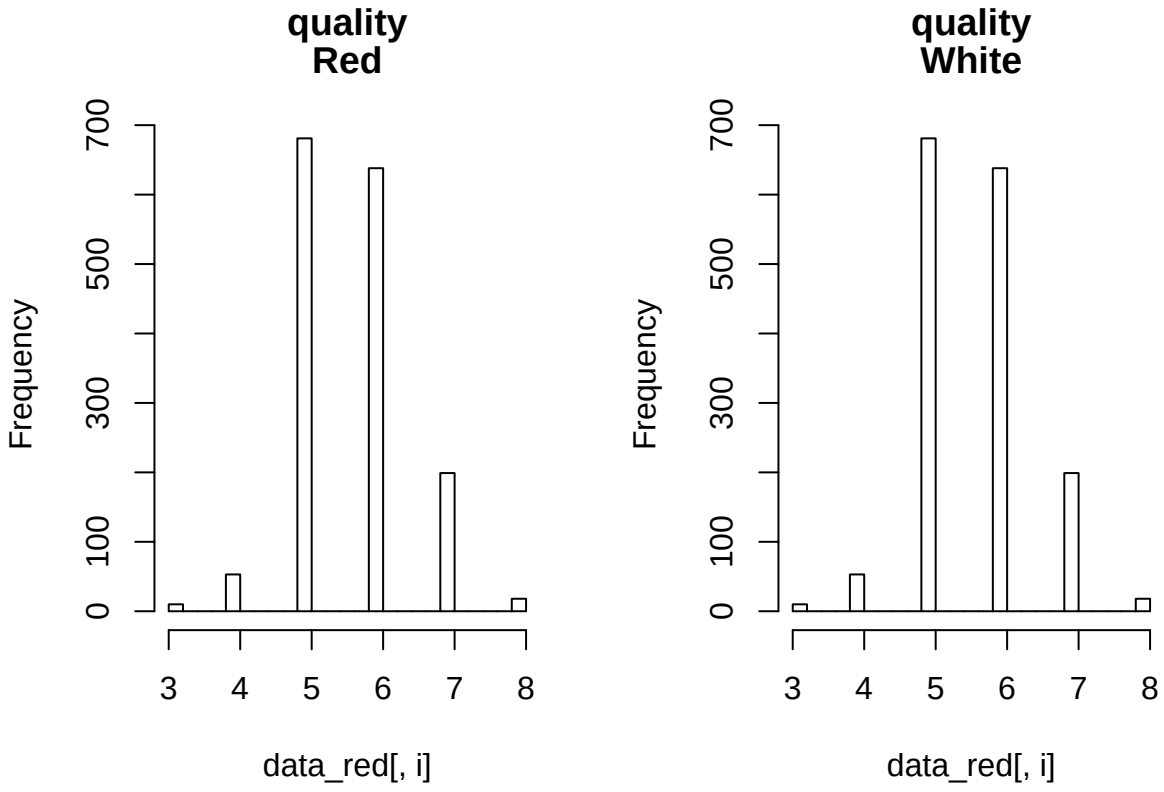

pH
White
Frequency
0
50
100
150
200
2.8
3.2
3.6
4.0
data_red[, i]


sulphates
Red
Frequency
0
50
100
200
300
0.5
1.0
1.5
2.0
data_red[, i]


sulphates
White
Frequency
0
50
100
200
300
0.5
1.0
1.5
2.0
data_red[, i]


alcohol
Red
Frequency
0
50
100
150
200
9
10
12
14
data_red[, i]


alcohol
White
Frequency
0
50
100
150
200
9
10
12
14
data_red[, i]



Aspectos destacables del análisis preliminar previa limpieza:

- Todas las variables son numéricas de tipo float, excepto el valor de calidad que es un integer, y la variable style que es un factor.
- Como se puede ver en el summary, la escala de las variables oscila entre variables con valores <1 y variables con valores de >200, a causa de esta diferencia de escala será interesante plantearse llevar a cabo algún tipo de normalización.
- Los boxplots muestran diversas variables con outliers, así como variables que parecen significativamente diferentes entre vinos blancos y tintos.
- El número de vinos blancos analizados es más del triple que el de vinos tintos.
- En algunas variables, una vez vistos los histogramas, podemos sospechar falta de normalidad.

3 Limpieza del dataset.

3.1 Integración de los datos.

En este dataset, la integración ya ha sido llevada a cabo. El dataset "Wine Quality Dataset", se encuentra disponible en el repositorio UCI: Wine Quality Dataset, y consta de dos datasets separados, el "winequality-white.csv" y el "winequality-red.csv", Kaggle importó estos dataset y llevó a cabo la integración vertical de ambos, añadiendo una variable categórica "style" que permite diferenciar los vinos blancos de los tintos. Por tanto para este dataset no se llevarán a cabo tareas de integración de datos.

3.2 Selección de datos de interés.

Una vez realizado el análisis descriptivo previo, podemos ver que todas las variables del dataset son importantes, en mayor o menor medida, para los análisis que realizaremos más adelante, por lo que no podemos prescindir de ninguna y todas las variables serán seleccionadas para el dataset de trabajo.

3.3 Reducción.

En este dataset NO se llevará a cabo una reducción de la cantidad de registros disponibles, ya que tampoco hay una cantidad excesiva de los mismos (unos 6500 registros) que genere tiempos de computación, ni dificultades excesivas en su procesado. Eliminar registros implicaría para este caso perder información valiosa, por lo que no se llevará a cabo esta eliminación.

En cuanto a una posible reducción previa de dimensionalidad, realizaremos un análisis PCA previo, con las variables físico químicas exclusivamente:

```
data.pca <- prcomp(data[,c(1:11)], center = TRUE, scale = TRUE)
summary(data.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7407 1.5792 1.2475 0.98517 0.84845 0.77930 0.72330
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521 0.04756
## Cumulative Proportion  0.2754 0.5021 0.6436 0.73187 0.79732 0.85253 0.90009
##                           PC8     PC9   PC10    PC11
## Standard deviation     0.70817 0.58054 0.4772 0.18119
## Proportion of Variance 0.04559 0.03064 0.0207 0.00298
## Cumulative Proportion  0.94568 0.97632 0.9970 1.00000
```

Como podemos ver, necesitamos entre 6 y 7 componentes principales para acumular un 90% de variancia, por lo que no es viable reducir el número de dimensiones inicial, a un número de dimensiones más manejable (tres o cuatro).

Más adelante, cuando empleemos modelos de regresión, podremos ver si alguna de las variables físico químicas tiene un peso específico poco relevante en el modelo.

3.4 Conversión.

3.4.1 Normalización.

Tal como ya se ha comentado, hay variables que difieren entre si al menos tres o cuatro órdenes de magnitud (por ejemplo los cloruros con una media entorno a 0.05 mg/l y el total de SO2 con una media de 115 mg/l), por lo que considero mecesario estandarizar los datos para eliminar el sesgo que puede introducir esta disparidad de escalas. La estandarización empleada sería la z-score.

```
data[,c(1:11)]<- scale(data[,c(1:11)])
#summary(data)

#Estandarizamos también data_red y data_white, good y bad ya están estandarizados.
data_red[,c(1:11)]<- scale(data_red[,c(1:11)])
data_white[,c(1:11)]<- scale(data_white[,c(1:11)])
```

Podemos ver que hay variables adquieren valores 5, 6, 7 y hasta 15 desviaciones estándar respecto de su media, sobretodo en los valores mayores. En los valores menores no se pasa de las tres desviaciones estándar, lo cual es mucho más razonable.

3.4.2 Transformación de Box-Cox.

Por ahora no decidiremos si empleamos este tipo de transformación, más adelante en el apartado de análisis estadístico inferencial decidiremos si realizamos esta transformación para alguna de las variables del dataset, o bien empleamos tests no paramétricos para realizar comparaciones.

3.4.3 Discretización.

En este dataset puede ser interesante discretizar la variable "quality" en una variable dicotómica, ya que esto nos permitiría realizar una regresión logística, o bien emplear otro tipo de clasificadores en combinación con curvas ROC como medidas de rendimiento.

LA discretización que llevaremos a cabo será clasificar los vinos con un valor de calidad mayor o igual a 7 como vinos de buena calidad y los de valor menor de 7 como vinos normales o malos.

```
data$class[data$quality >= 7]<- "good"
data$class[data$quality < 7]<- "bad"
data$class <- as.factor(data$class)

str(data$class)
```

```
##  Factor w/ 2 levels "bad","good": 1 1 1 1 1 1 1 2 2 1 ...
```

```
#Separamos también los datasets, por si más adelante es necesario utilizarlos.
data_good <- data[which(data$class == "good"), ]
data_bad <- data[data$class == "bad", ]

#Exportamos el dataset de trabajo a CSV.
write.csv(data, file="wine_dataset_PRAC2.csv", row.names = FALSE)
```

3.5 Ceros y datos vacíos.

En este dataset, dado que contiene variables físico químicas numéricas, los valores cero son posibles y tienen sentido físico, por ejemplo para el ácido cítrico, existen varios valores cero, [ácido cítrico] = 0 mgr/l, lo cual significa que no contiene este compuesto, o bien que su valor está por debajo del límite de detección del método analítico. Por lo tanto NO consideraremos los valores cero en las variables numéricas (antes de normalizar) como valores anómalos y los dejaremos tal como están.

Podrían ser anómalos algunos valores cero como por ejemplo en el caso de la densidad, pero como podemos comprobar en el summary(data), en todas las variables físico químicas numéricas el valor mínimo es mayor que cero, incluida la densidad, excepto para el caso del ácido cítrico, para el cual ya hemos comentado que estos valores no pueden ser considerados anormales, y por tanto hay que mantenerlos.

Los datos vacíos o NA, son otro asunto que hay que comprobar previamente, para ello:

```
# Estadísticas de valores NA
colSums(is.na(data))
```

```
##        fixed_acidity     volatile_acidity          citric_acid
##                    0                    0                    0
##       residual_sugar            chlorides  free_sulfur_dioxide
##                    0                    0                    0
## total_sulfur_dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
##                style                class
##                    0                    0
```

```
# Atributos con valor ausente, datos vacíos.
colSums(data=="")
```

```
##        fixed_acidity     volatile_acidity          citric_acid
##                    0                    0                    0
##       residual_sugar            chlorides  free_sulfur_dioxide
```

```
##                    0                    0                    0
## total_sulfur_dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
##                style                class
##                    0                    0
```

Como podemos comprobar no existen en este dataset NA o valores vacíos.

En el caso de que se hubiesen detectado valores cero anormales (por ejemplo en la densidad), o valores NA o vacíos, para este dataset lo más aconsejable sería sustituir los mismos por valores estimados, por ejemplo empleando métodos de similitud como el kNN-imputation (donse se buscan los registros más similares por distancia al registro que contiene el NA y se le asigna la media de este clúster para el atributo), que considero son mejores que sustituir el valor NA por la media del atributo en el dataset. Estos métodos siempre introducen un cierto sesgo, el que menos quizás el KNN-imputation, pero es preferible introducir este sesgo, a perder la información de los registros que contienen estos NA. En el caso de que el número de registros que contienen NA fuese muy reducido, podrísmod plsntearnos eliminarlos.

3.6 Tratamiento de valores extremos.

Los outliers son valores que se encuentran muy alejados de la distribución normal de una variable, normalmente se emplean 3 desviaciones estándar como referencia para clasificar un valor extremo. Como hemos visto al normalizar las variables del dataset, en todas ellas aparecen outliers, con valores que incluso llegan a 15 desviaciones estándar en algunas de ellas, la mayoría de estos outliers como hemos visto se encuentran en el rango superior de las variables, no así en los inferiores.

A priori, los outliers observados son legítimos y por tanto considero hay que mantenerlos en el análisis. Una posible explicación sería que la producción de vinos es un proceso biológico con un importante componente artesanal y fuertemente dependiente de la climatología y de geografía de los cultivos, por lo que es fácil que en los parámetros analizados, aparezcan fuertes desviaciones de la normalidad en función del año y la climatología.

En particular si analizamos el dataset conjunto (blancos y tintos) y los datasets de vino tinto y blanco por separado, podemos observar que en el dataset mixto (blancos y tintos), el número de outliers se dispara, lo cual ya nos indica que la diferencia existente entre ambos tipos de vinos es posiblemente significativa.

Si analizamos en detalle el número de outliers en cada variables, tenemos los siguientes valores:

Variable	outliers dataset blancos+tintos	outliers dataset blancos	outliers dataset tintos
fixed_acidity	357	137	49
volatile_acidity	377	186	19
citric_acid	509	270	1
residual_sugar	115	7	155
chlorides	286	208	112
free_SO2	62	50	33
total_SO2	10	19	55
density	3	5	45
pH	73	75	35
sulphates	191	124	59
alcohol	3	0	14

La presencia de outliers tan extremos podría generar problemas en los análisis estadísticos. Una posible opción para corregir parcialmente la distorsión que introducen estos outliers, dado que hemos normalizado

los datos de los datasets mediante una estandarización z-score, es sustituir todo valor mayor de 3 o menor de -3 (valores con más de 3 veces la desviación estándar de la media de la variable), por 3 o -3 respectivamente. Mediante esta transformación, los valores modificados continúan siendo outliers (ya que su sd es 3, -3), pero se consiguen eliminar valores extremos de sd, por ejemplo 9, 15, etc.

El código de R para realizar esta transformación es el siguiente:

```
for (i in 1:11) {
  for (j in 1: (nrow(data))) {
    if (data[j, i] > 3){
      data[j, i] <- 3
    }
    if (data[j, i] < -3){
      data[j, i] <- -3
    }
  }
}

for (i in 1:11) {
  for (j in 1: (nrow(data_red))) {
    if (data_red[j, i] > 3){
      data_red[j, i] <- 3
    }
    if (data_red[j, i] < -3){
      data_red[j, i] <- -3
    }
  }
}

for (i in 1:11) {
  for (j in 1: (nrow(data_white))) {
    if (data_white[j, i] > 3){
      data_white[j, i] <- 3
    }
    if (data_white[j, i] < -3){
      data_white[j, i] <- -3
    }
  }
}
```

A pesar de ello tras realizar la transformación indicada, se observa que los tests estadísticos del apartado 4.3 y 4.4 continúan dando los mismos resultados, no varían en absoluto, y que de los análisis realizados tampoco ninguno mejora sus medidas de rendimiento.

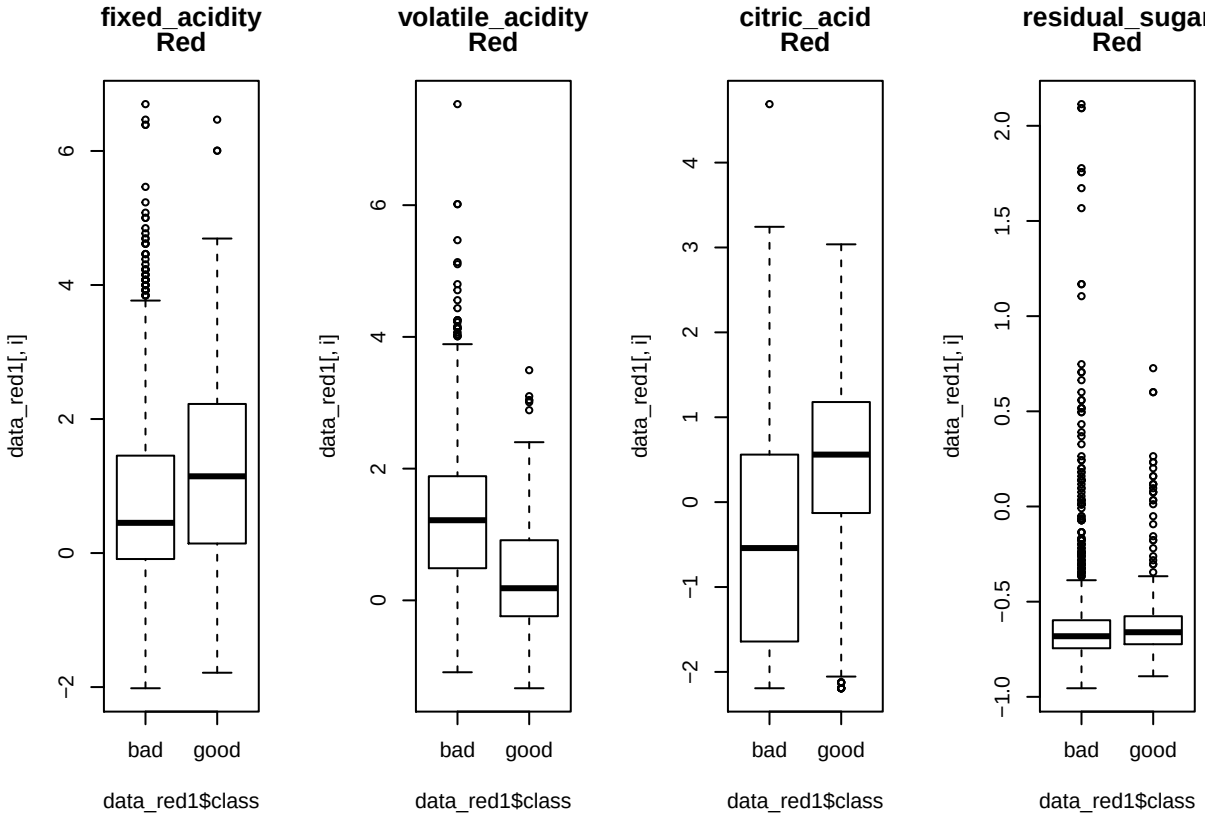
Por tanto vistos los resultados de esta transformación de los outliers, **NO** la llevaremos a cabo, ya que no introduce ninguna mejora, por lo que dejaremos los outliers tal como están sin eliminar ni modificar ninguno.

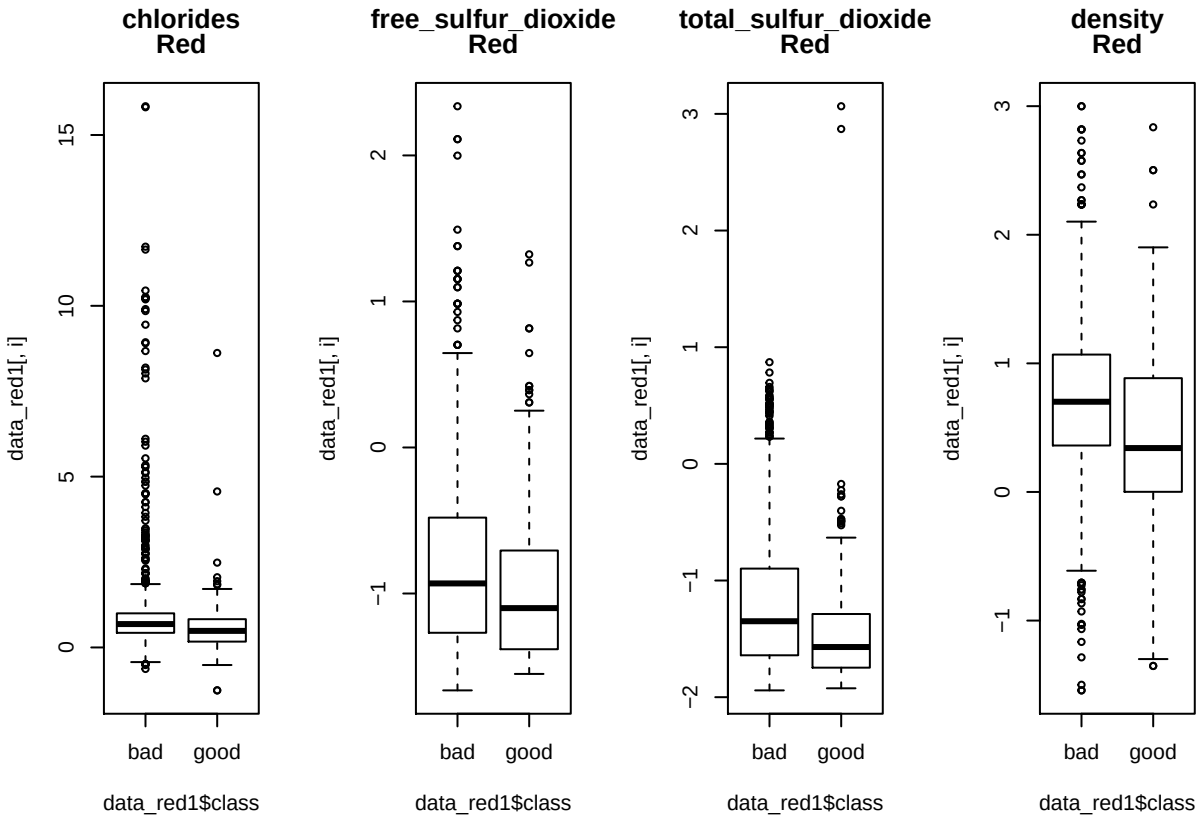
4 Análisis de los datos.

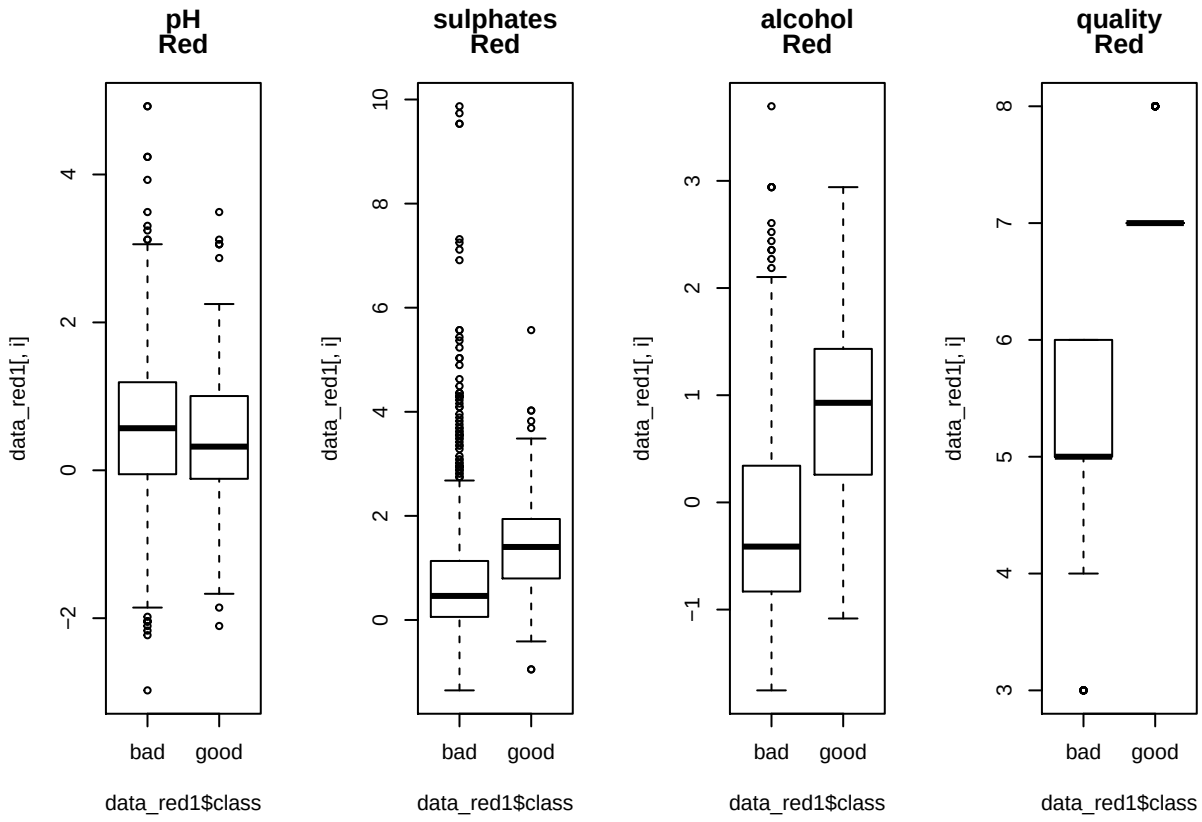
4.1 Análisis descriptivo.

Ya ha sido llevado a cabo en etapas anteriores, dado que el dataset inicial no se ha modificado en exceso, solamente se añadirá al análisis descriptivo inicial (apartado 2.3), los boxplots para comparar las variables químico físicas frente frente a la nueva categoría que hemos añadido “good”, “bad”.

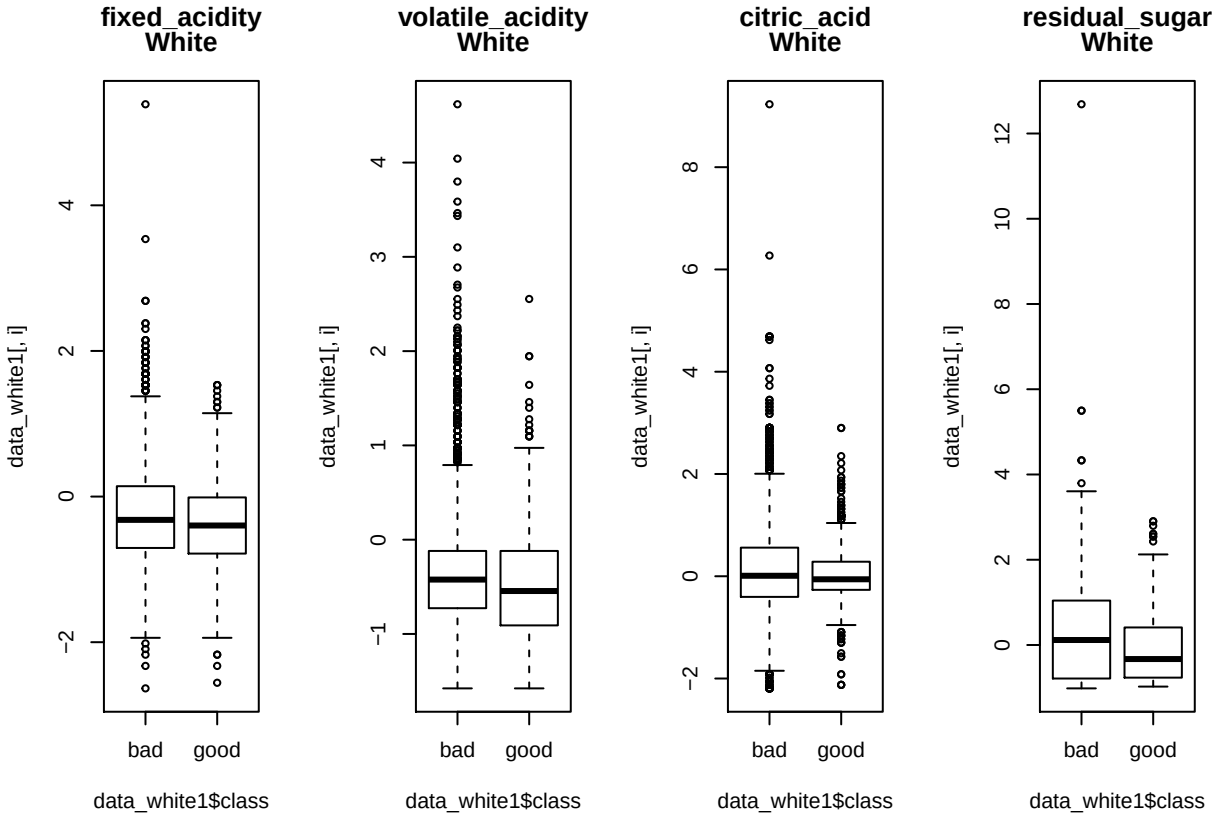
```r
data_red1 <- data[which(data$style == "red"), ]
data_white1 <- data[data$style == "white", ]

#Generamos boxplots de las variables separando en vinos de buena calidad y mala calidad, para tintos y
par(mfrow=c(1,4))
for (i in 1:(ncol(data_red1)-2)) {
    boxplot(data_red1[,i] ~ data_red1$class,
    main=c(names(data_red1[i]), "Red"))
}
```

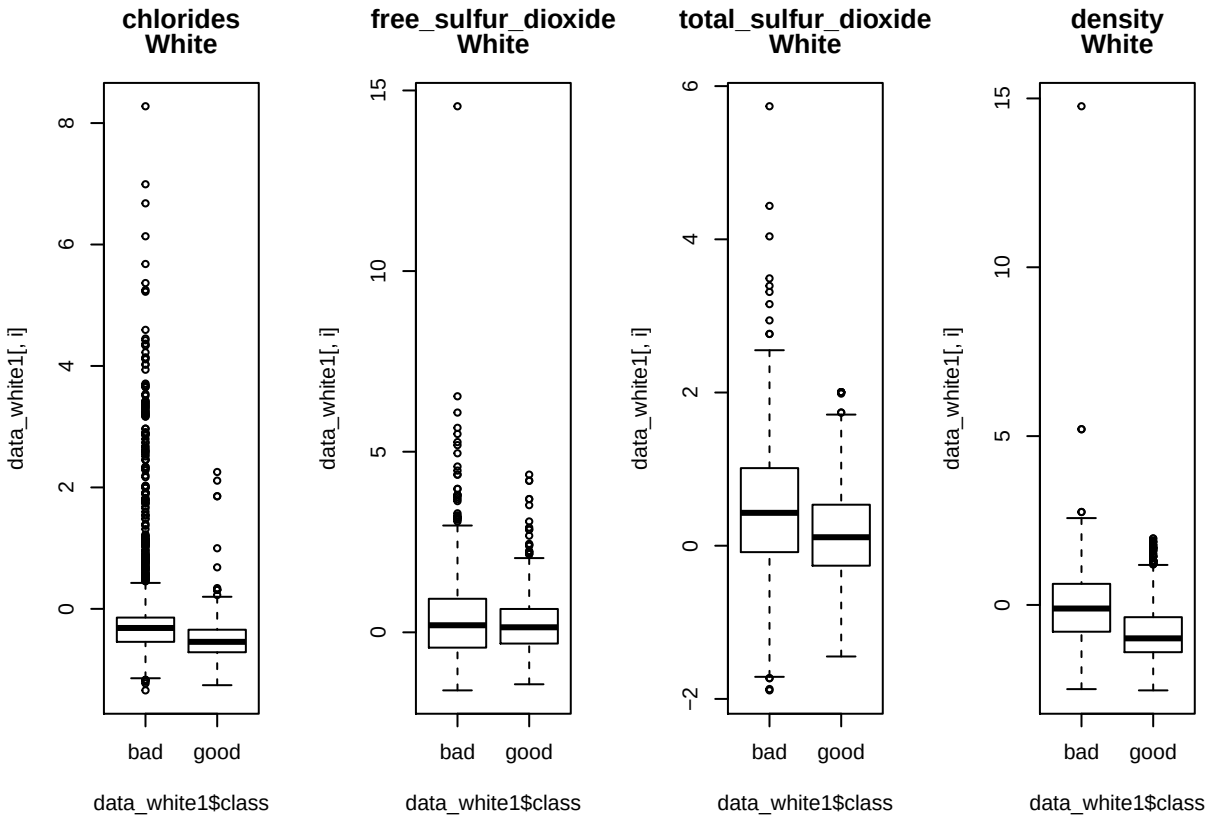


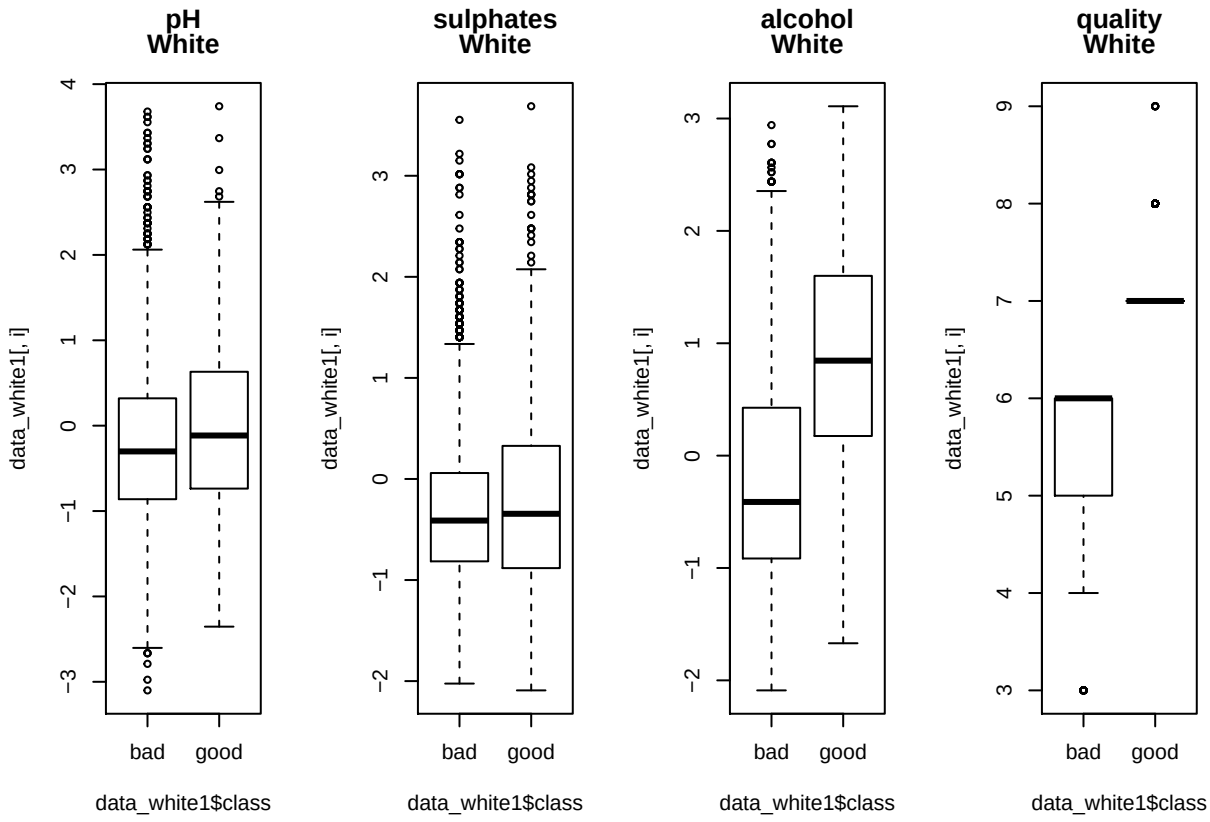

chlorides
Red
data_red1[, i]
0
5
10
15
bad
good
data_red1$class
free_sulfur_dioxide
Red
data_red1[, i]
−1
0
1
2
bad
good
data_red1$class
total_sulfur_dioxide
Red
data_red1[, i]
−2
−1
0
1
2
3
bad
good
data_red1$class
density
Red
data_red1[, i]
−1
0
1
2
3
bad
good
data_red1$class




```
par(mfrow=c(1,4))
for (i in 1:(ncol(data_white1)-2)) {
    boxplot(data_white1[,i] ~ data_white1$class,
    main=c(names(data_white1[i]), "White"))
}
```


fixed_acidity
White
data_white1[, i]
data_white1$class
bad
good
volatile_acidity
White
citric_acid
White
residual_sugar
White


chlorides
White
data_white1[, i]
0
2
4
6
8
bad
good
data_white1$class
free_sulfur_dioxide
White
data_white1[, i]
0
5
10
15
bad
good
data_white1$class
total_sulfur_dioxide
White
data_white1[, i]
−2
0
2
4
6
bad
good
data_white1$class
density
White
data_white1[, i]
0
5
10
15
bad
good
data_white1$class



A primera vista (antes de realizar los tests estadísticos), las variables que parece que afectan más a la calidad excelente de un vino (class >= 7) en los boxplots (mayor diferencia entre "good" y "bad") son:

- Para los tintos alcohol, sulphates, total_SO2, volatile_acidity, density y citric_acid.
- Para los blancos alcohol y density.

4.2 Selección de los grupos de datos que se quieren comparar.

Básicamente compararemos si existen diferencias significativas entre las variables del dataset dentro de los siguientes grupos:

- Vinos blancos y tintos.
- Vinos de buena calidad y de mala calidad.

Una vez vistas las diferencias significativas entre blancos y tintos si es que las hay, analizaremos si es posible realizar predicciones de calidad de los vinos mediante regresiones. Estas regresiones nos permitirán además ver si alguno de los parámetros físico químicos tiene un mayor impacto en la calidad final del vino.

En concreto realizaremos:

- Regresión lineal.

Finalmente crearemos modelos clasificadores para intentar predecir la calidad de un vino en base a sus propiedades físico químicas, mediante:

- Regresión logística.
- Random Forest.

4.3 Comprobación de normalidad y homocedasticidad de las variables.

4.3.1 fixed_acidity.

```
#Test de normalidad red-white.
shapiro.test(data_red$fixed_acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$fixed_acidity
## W = 0.94203, p-value < 2.2e-16
```

```
shapiro.test(data_white$fixed_acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$fixed_acidity
## W = 0.97656, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(fixed_acidity ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  fixed_acidity by style
## Fligner-Killeen:med chi-squared = 747.29, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$fixed_acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$fixed_acidity
## W = 0.8575, p-value < 2.2e-16
```

```
#El shapiro test solo admite máximo 5000 registros por lo que hay que hacer un random sample de la mues
shapiro.test(sample(data_bad$fixed_acidity, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$fixed_acidity, 5000, replace = FALSE)
## W = 0.88123, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(fixed_acidity ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
```

```
## data:  fixed_acidity by class
## Fligner-Killeen:med chi-squared = 0.72109, df = 1, p-value = 0.3958
```

Los dos grupos **SI** son homocedásticos.

4.3.2 volatile_acidity.

```
#Test de normalidad red-white.
shapiro.test(data_red$volatile_acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$volatile_acidity
## W = 0.97434, p-value = 2.693e-16
```

```
shapiro.test(data_white$volatile_acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$volatile_acidity
## W = 0.90455, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(volatile_acidity ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  volatile_acidity by style
## Fligner-Killeen:med chi-squared = 842.58, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$volatile_acidity)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$volatile_acidity
## W = 0.91914, p-value < 2.2e-16
```

```
shapiro.test(sample(data_bad$volatile_acidity, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$volatile_acidity, 5000, replace = FALSE)
## W = 0.87716, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(volatile_acidity ~ class, data= data)
```

```
##
```

```
##  Fligner-Killeen test of homogeneity of variances
##
## data:  volatile_acidity by class
## Fligner-Killeen:med chi-squared = 73.358, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

4.3.3 citric_acid.

```
#Test de normalidad red-white.
shapiro.test(data_red$citric_acid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$citric_acid
## W = 0.95529, p-value < 2.2e-16
```

```
shapiro.test(data_white$citric_acid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$citric_acid
## W = 0.92225, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(citric_acid ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  citric_acid by style
## Fligner-Killeen:med chi-squared = 842.68, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$citric_acid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$citric_acid
## W = 0.9308, p-value < 2.2e-16
```

```
shapiro.test(sample(data_bad$citric_acid, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$citric_acid, 5000, replace = FALSE)
## W = 0.96835, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(citric_acid ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  citric_acid by class
## Fligner-Killeen:med chi-squared = 145.55, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

4.3.4 residual_sugar.

```
#Test de normalidad red-white.
shapiro.test(data_red$residual_sugar)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$residual_sugar
## W = 0.56608, p-value < 2.2e-16
```

```
shapiro.test(data_white$residual_sugar)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$residual_sugar
## W = 0.88457, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(residual_sugar ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  residual_sugar by style
## Fligner-Killeen:med chi-squared = 1766.1, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$residual_sugar)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$residual_sugar
## W = 0.8148, p-value < 2.2e-16
```

```
shapiro.test(sample(data_bad$residual_sugar, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$residual_sugar, 5000, replace = FALSE)
## W = 0.82613, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```r
#Test de homocedasticidad good-bad.
fligner.test(residual_sugar ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  residual_sugar by class
## Fligner-Killeen:med chi-squared = 49.482, df = 1, p-value = 2.002e-12
```

Los dos grupos **NO** son homocedásticos.

4.3.5 chlorides.

```r
#Test de normalidad red-white.
shapiro.test(data_red$chlorides)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$chlorides
## W = 0.48425, p-value < 2.2e-16
```

```r
shapiro.test(data_white$chlorides)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$chlorides
## W = 0.59081, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```r
#Test de homocedasticidad red-white.
fligner.test(chlorides ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  chlorides by style
## Fligner-Killeen:med chi-squared = 203.81, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```r
#Test de normalidad good-bad.
shapiro.test(data_good$chlorides)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$chlorides
## W = 0.74784, p-value < 2.2e-16
```

```r
shapiro.test(sample(data_bad$chlorides, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$chlorides, 5000, replace = FALSE)
```

```
## W = 0.61301, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(chlorides ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  chlorides by class
## Fligner-Killeen:med chi-squared = 88.495, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

4.3.6 free_sulfur_dioxide.

```
#Test de normalidad red-white.
shapiro.test(data_red$free_sulfur_dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$free_sulfur_dioxide
## W = 0.90184, p-value < 2.2e-16
```

```
shapiro.test(data_white$free_sulfur_dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$free_sulfur_dioxide
## W = 0.94207, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(free_sulfur_dioxide ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  free_sulfur_dioxide by style
## Fligner-Killeen:med chi-squared = 345.11, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$free_sulfur_dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$free_sulfur_dioxide
## W = 0.96957, p-value = 9.634e-16
```

```
shapiro.test(sample(data_bad$free_sulfur_dioxide, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  sample(data_bad$free_sulfur_dioxide, 5000, replace = FALSE)
## W = 0.93235, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(free_sulfur_dioxide ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  free_sulfur_dioxide by class
## Fligner-Killeen:med chi-squared = 59.234, df = 1, p-value = 1.4e-14
```

Los dos grupos **NO** son homocedásticos.

4.3.7 total_sulfur_dioxide.

```
#Test de normalidad red-white.
shapiro.test(data_red$total_sulfur_dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$total_sulfur_dioxide
## W = 0.87322, p-value < 2.2e-16
```

```
shapiro.test(data_white$total_sulfur_dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$total_sulfur_dioxide
## W = 0.98901, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(total_sulfur_dioxide ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  total_sulfur_dioxide by style
## Fligner-Killeen:med chi-squared = 199.15, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$total_sulfur_dioxide)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$total_sulfur_dioxide
## W = 0.97168, p-value = 4.035e-15
```

```
shapiro.test(sample(data_bad$total_sulfur_dioxide, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$total_sulfur_dioxide, 5000, replace = FALSE)
## W = 0.98068, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(total_sulfur_dioxide ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  total_sulfur_dioxide by class
## Fligner-Killeen:med chi-squared = 121.2, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

4.3.8 density.

```
#Test de normalidad red-white.
shapiro.test(data_red$density)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$density
## W = 0.99087, p-value = 1.936e-08
```

```
shapiro.test(data_white$density)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$density
## W = 0.9548, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(density ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  density by style
## Fligner-Killeen:med chi-squared = 449.05, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$density)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$density
## W = 0.94682, p-value < 2.2e-16
```

```r
shapiro.test(sample(data_bad$density, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$density, 5000, replace = FALSE)
## W = 0.963, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```r
#Test de homocedasticidad good-bad.
fligner.test(density ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  density by class
## Fligner-Killeen:med chi-squared = 7.9727, df = 1, p-value = 0.004749
```

Los dos grupos **NO** son homocedásticos.

4.3.9 pH.

```r
#Test de normalidad red-white.
shapiro.test(data_red$pH)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$pH
## W = 0.99349, p-value = 1.712e-06
```

```r
shapiro.test(data_white$pH)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$pH
## W = 0.9881, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```r
#Test de homocedasticidad red-white.
fligner.test(pH ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pH by style
## Fligner-Killeen:med chi-squared = 0.40179, df = 1, p-value = 0.5262
```

Los dos grupos **SI** son homocedásticos.

```r
#Test de normalidad good-bad.
shapiro.test(data_good$pH)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  data_good$pH
## W = 0.993, p-value = 9.693e-06
```

```
shapiro.test(sample(data_bad$pH, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$pH, 5000, replace = FALSE)
## W = 0.99, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(pH ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pH by class
## Fligner-Killeen:med chi-squared = 0.452, df = 1, p-value = 0.5014
```

Los dos grupos **SI** son homocedásticos.

4.3.10 sulphates.

```
#Test de normalidad red-white.
shapiro.test(data_red$sulphates)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$sulphates
## W = 0.83304, p-value < 2.2e-16
```

```
shapiro.test(data_white$sulphates)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$sulphates
## W = 0.95161, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(sulphates ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  sulphates by style
## Fligner-Killeen:med chi-squared = 82.934, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$sulphates)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  data_good$sulphates
## W = 0.95199, p-value < 2.2e-16
```

```
shapiro.test(sample(data_bad$sulphates, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$sulphates, 5000, replace = FALSE)
## W = 0.87564, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(sulphates ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  sulphates by class
## Fligner-Killeen:med chi-squared = 94.301, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

4.3.11 alcohol.

```
#Test de normalidad red-white.
shapiro.test(data_red$alcohol)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$alcohol
## W = 0.92884, p-value < 2.2e-16
```

```
shapiro.test(data_white$alcohol)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$alcohol
## W = 0.9553, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(alcohol ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  alcohol by style
## Fligner-Killeen:med chi-squared = 78.704, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

```
#Test de normalidad good-bad.
shapiro.test(data_good$alcohol)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$alcohol
## W = 0.97656, p-value = 1.508e-13
```

```
shapiro.test(sample(data_bad$alcohol, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$alcohol, 5000, replace = FALSE)
## W = 0.94282, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad good-bad.
fligner.test(alcohol ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  alcohol by class
## Fligner-Killeen:med chi-squared = 37.784, df = 1, p-value = 7.904e-10
```

Los dos grupos **NO** son homocedásticos.

4.3.12 quality.

```
#Test de normalidad red-white.
shapiro.test(data_red$quality)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_red$quality
## W = 0.85759, p-value < 2.2e-16
```

```
shapiro.test(data_white$quality)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_white$quality
## W = 0.88904, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```
#Test de homocedasticidad red-white.
fligner.test(quality ~ style, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  quality by style
## Fligner-Killeen:med chi-squared = 0.61775, df = 1, p-value = 0.4319
```

Los dos grupos **SI** son homocedásticos.

```r
#Test de normalidad good-bad.
shapiro.test(data_good$quality)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_good$quality
## W = 0.44155, p-value < 2.2e-16
```

```r
shapiro.test(sample(data_bad$quality, 5000, replace=FALSE))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(data_bad$quality, 5000, replace = FALSE)
## W = 0.71186, p-value < 2.2e-16
```

Los datos **NO** se distribuyen normalmente.

```r
#Test de homocedasticidad good-bad.
fligner.test(quality ~ class, data= data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  quality by class
## Fligner-Killeen:med chi-squared = 356.46, df = 1, p-value < 2.2e-16
```

Los dos grupos **NO** son homocedásticos.

4.3.13 Resumen.

Variable	Normalidad Red	Normalidad White	Homocedasticidad style	Homodedaticidad class
fixed_acidity	NO	NO	NO	SI
volatile_acidity	NO	NO	NO	NO
citric_acid	NO	NO	NO	NO
residual_sugar	NO	NO	NO	NO
chlorides	NO	NO	NO	NO
free_SO2	NO	NO	NO	NO
total_SO2	NO	NO	NO	NO
density	NO	NO	NO	NO
pH	NO	NO	SI	SI
sulphates	NO	NO	NO	NO
alcohol	NO	NO	NO	NO
quality	NO	NO	SI	NO

4.4 Pruebas estadísticas de comparación de variables.

Dado que en ninguna de las variables (tanto en red-white como en good-bad) tenemos normalidad y en la mayoría tampoco tenemos homocedasticidad, emplearemos el test no paramétrico de Wilcoxon o el de Mann-Whitney que se aplican indistintamente mediante la función wilcos.test().

4.4.1 fixed_acidity.

```
#Comparación red-white.
wilcox.test(fixed_acidity ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  fixed_acidity by style
## W = 6138507, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(fixed_acidity ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  fixed_acidity by class
## W = 118472, p-value = 6.4e-07
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(fixed_acidity ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  fixed_acidity by class
## W = 2207934, p-value = 1.973e-05
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.2 volatile_acidity.

```
#Comparación red-white
wilcox.test(volatile_acidity ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  volatile_acidity by style
## W = 7059624, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(volatile_acidity ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  volatile_acidity by class
```

```
## W = 223449, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(volatile_acidity ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  volatile_acidity by class
## W = 2223712, p-value = 3.245e-06
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.3 citric_acid.

```
#Comparación red-white
wilcox.test(citric_acid ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  citric_acid by style
## W = 3070089, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(citric_acid ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  citric_acid by class
## W = 96501, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(citric_acid ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  citric_acid by class
## W = 2037318, p-value = 0.9378
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **NO** son estadísticamente significativas. Por tanto la concentración de ácido citrico de los dos tipos de vinos (good/bad) se puede considerar similar.

4.4.4 residual_sugar.

```
#Comparación red-white
wilcox.test(residual_sugar ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  residual_sugar by style
## W = 2569687, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(residual_sugar ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  residual_sugar by class
## W = 134725, p-value = 0.01587
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(residual_sugar ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  residual_sugar by class
## W = 2289208, p-value = 3.86e-10
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.5 chlorides.

```
#Comparación red-white
wilcox.test(chlorides ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  chlorides by style
## W = 7407016, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(chlorides ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  chlorides by class
```

```
## W = 189611, p-value = 3.523e-10
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(chlorides ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  chlorides by class
## W = 2808906, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.6 free_sulfur_dioxide.

```
#Comparación red-white.
wilcox.test(free_sulfur_dioxide ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  free_sulfur_dioxide by style
## W = 1186397, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(free_sulfur_dioxide ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  free_sulfur_dioxide by class
## W = 172440, p-value = 0.0003705
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(free_sulfur_dioxide ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  free_sulfur_dioxide by class
## W = 2067264, p-value = 0.4163
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **NO** son estadísticamente significativas. Por tanto la concentración libre de SO2 de los dos tipos de vinos (good/bad) se puede considerar similar.

4.4.7 total_sulfur_dioxide.

```
#Comparación red-white
wilcox.test(total_sulfur_dioxide ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  total_sulfur_dioxide by style
## W = 366640, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(total_sulfur_dioxide ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  total_sulfur_dioxide by class
## W = 193409, p-value = 6.255e-12
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(total_sulfur_dioxide ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  total_sulfur_dioxide by class
## W = 2513178, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.8 density.

```
#Comparación red-white
wilcox.test(density ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  density by style
## W = 6059285, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(density ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  density by class
```

```
## W = 188154, p-value = 1.52e-09
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(density ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  density by class
## W = 2904671, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.9 pH.

```
#Comparación red-white
wilcox.test(pH ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pH by style
## W = 5681840, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(pH ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pH by class
## W = 166767, p-value = 0.007804
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(pH ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pH by class
## W = 1763211, p-value = 2.941e-11
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.10 sulphates.

```
#Comparación red-white
wilcox.test(sulphates ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  sulphates by style
## W = 6509961, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad red wines.
wilcox.test(sulphates ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  sulphates by class
## W = 78323, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(sulphates ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  sulphates by class
## W = 1971898, p-value = 0.1265
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **NO** son estadísticamente significativas. Por tanto la concentración de sulfatos en los dos tipos de vinos (good/bad) se puede considerar similar.

4.4.11 alcohol.

```
#Comparación red-white
wilcox.test(alcohol ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  alcohol by style
## W = 3829044, p-value = 0.1818
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **NO** son estadísticamente significativas. Por tanto la graduación alcohólica de los dos tipos de vinos se puede considerar similar.

```
#Comparación good-bad red wines.
wilcox.test(alcohol ~ class, data = data_red1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  alcohol by class
## W = 53234, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

```
#Comparación good-bad white wines.
wilcox.test(alcohol ~ class, data = data_white1)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  alcohol by class
## W = 1007694, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.12 quality.

```
#Comparación red-white
wilcox.test(quality ~ style, data = data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  quality by style
## W = 3311514, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Las diferencias entre los dos grupos al respecto de esta variable **SI** son estadísticamente significativas.

4.4.13 Resumen.

Variable	Diferencias significativas entre tintos y blancos (style Red-White)	Diferencias significativas vinos tintos entre buenos y malos (class Good-Bad)	Diferencias significativas vinos blancos entre buenos y malos (class Good-Bad)
fixed_acidity	SI	SI	SI
volatile_acidity	SI	SI	SI
citric_acid	SI	SI	NO
residual_sugar	SI	SI	SI
chlorides	SI	SI	SI
free_SO2	SI	SI	NO
total_SO2	SI	SI	SI
density	SI	SI	SI
pH	SI	SI	SI
sulphates	SI	SI	NO
alcohol	NO	SI	SI
quality	SI		

4.5 Correlación entre variables.

Dado que las distribuciones de las variables no son normales empleamos la correlación de Spearman:

```
#Vinos tintos.
cor(data_red[,-c(13,14)], method= "spearman")
```

```
##                      fixed_acidity volatile_acidity  citric_acid residual_sugar
## fixed_acidity          1.00000000      -0.27828222  0.661708417     0.22070086
```

```
## volatile_acidity       -0.27828222       1.00000000 -0.610259467     0.03238560
## citric_acid             0.66170842      -0.61025947  1.000000000     0.17641731
## residual_sugar          0.22070086       0.03238560  0.176417306     1.00000000
## chlorides               0.25090411       0.15877025  0.112576508     0.21295924
## free_sulfur_dioxide    -0.17513656       0.02116264 -0.076451575     0.07461786
## total_sulfur_dioxide   -0.08841741       0.09411014  0.009399602     0.14537506
## density                 0.62307076       0.02501412  0.352285261     0.42226586
## pH                     -0.70667359       0.23357152 -0.548026276    -0.08997095
## sulphates               0.21265375      -0.32558398  0.331074404     0.03833200
## alcohol                -0.06657566      -0.22493168  0.096455544     0.11654813
## quality                 0.11408367      -0.38064651  0.213480914     0.03204817
##                          chlorides free_sulfur_dioxide total_sulfur_dioxide
## fixed_acidity         0.2509041064       -0.1751365613        -0.0884174083
## volatile_acidity      0.1587702548        0.0211626414         0.0941101376
## citric_acid           0.1125765077       -0.0764515753         0.0093996024
## residual_sugar        0.2129592419        0.0746178640         0.1453750584
## chlorides             1.0000000000        0.0008051686         0.1300333418
## free_sulfur_dioxide   0.0008051686        1.0000000000         0.7896978767
## total_sulfur_dioxide  0.1300333418        0.7896978767         1.0000000000
## density               0.4113896972       -0.0411776800         0.1293321018
## pH                   -0.2343612736        0.1156791779        -0.0098414382
## sulphates             0.0208254792        0.0458623500        -0.0005038194
## alcohol              -0.2845039422       -0.0813673063        -0.2578060251
## quality              -0.1899223356       -0.0569006455        -0.1967350754
##                          density           pH     sulphates     alcohol
## fixed_acidity         0.62307076 -0.706673595  0.2126537506 -0.06657566
## volatile_acidity      0.02501412  0.233571519 -0.3255839818 -0.22493168
## citric_acid           0.35228526 -0.548026276  0.3310744040  0.09645554
## residual_sugar        0.42226586 -0.089970954  0.0383320002  0.11654813
## chlorides             0.41138970 -0.234361274  0.0208254792 -0.28450394
## free_sulfur_dioxide  -0.04117768  0.115679178  0.0458623500 -0.08136731
## total_sulfur_dioxide  0.12933210 -0.009841438 -0.0005038194 -0.25780603
## density               1.00000000 -0.312055078  0.1614782344 -0.46244458
## pH                   -0.31205508  1.000000000 -0.0803060380  0.17993243
## sulphates             0.16147823 -0.080306038  1.0000000000  0.20732955
## alcohol              -0.46244458  0.179932427  0.2073295535  1.00000000
## quality              -0.17707407 -0.043671935  0.3770601991  0.47853169
##                          quality
## fixed_acidity         0.11408367
## volatile_acidity     -0.38064651
## citric_acid           0.21348091
## residual_sugar        0.03204817
## chlorides            -0.18992234
## free_sulfur_dioxide  -0.05690065
## total_sulfur_dioxide -0.19673508
## density              -0.17707407
## pH                   -0.04367193
## sulphates             0.37706020
## alcohol               0.47853169
## quality               1.00000000
```

```
#Vinos blancos.
cor(data_white[,-c(13,14)], method= "spearman")
```

```
##                       fixed_acidity volatile_acidity citric_acid residual_sugar
```

```
## fixed_acidity           1.00000000     -0.042865228  0.29787793     0.10672494
## volatile_acidity       -0.04286523      1.000000000 -0.15040998     0.10862744
## citric_acid             0.29787793     -0.150409981  1.00000000     0.02462098
## residual_sugar          0.10672494      0.108627441  0.02462098     1.00000000
## chlorides               0.09469118     -0.004934156  0.03265950     0.22784390
## free_sulfur_dioxide    -0.02454223     -0.081212903  0.08831406     0.34610674
## total_sulfur_dioxide    0.11264866      0.117613959  0.09321867     0.43125248
## density                 0.27003091      0.010124341  0.09142519     0.78036485
## pH                     -0.41834116     -0.045203569 -0.14619267    -0.18002822
## sulphates              -0.01323781     -0.016902303  0.07976628    -0.00384398
## alcohol                -0.10682740      0.033966554 -0.02916996    -0.44525743
## quality                -0.08448545     -0.196561683  0.01833273    -0.08206979
##                        chlorides free_sulfur_dioxide total_sulfur_dioxide
## fixed_acidity         0.094691176        -0.024542230           0.11264866
## volatile_acidity     -0.004934156        -0.081212903           0.11761396
## citric_acid           0.032659495         0.088314056           0.09321867
## residual_sugar        0.227843904         0.346106737           0.43125248
## chlorides             1.000000000         0.167045505           0.37524367
## free_sulfur_dioxide   0.167045505         1.000000000           0.61861634
## total_sulfur_dioxide  0.375243666         0.618616339           1.00000000
## density               0.508301765         0.327821798           0.56382409
## pH                   -0.054006467        -0.006273578          -0.01182872
## sulphates             0.093930696         0.052251683           0.15782480
## alcohol              -0.570806407        -0.272569338          -0.47661933
## quality              -0.314488478         0.023713376          -0.19668029
##                           density           pH   sulphates     alcohol
## fixed_acidity          0.27003091 -0.418341158 -0.01323781 -0.10682740
## volatile_acidity       0.01012434 -0.045203569 -0.01690230  0.03396655
## citric_acid            0.09142519 -0.146192675  0.07976628 -0.02916996
## residual_sugar         0.78036485 -0.180028223 -0.00384398 -0.44525743
## chlorides              0.50830177 -0.054006467  0.09393070 -0.57080641
## free_sulfur_dioxide    0.32782180 -0.006273578  0.05225168 -0.27256934
## total_sulfur_dioxide   0.56382409 -0.011828718  0.15782480 -0.47661933
## density                1.00000000 -0.110060852  0.09507867 -0.82185508
## pH                    -0.11006085  1.000000000  0.14024331  0.14885725
## sulphates              0.09507867  0.140243305  1.00000000 -0.04486799
## alcohol               -0.82185508  0.148857249 -0.04486799  1.00000000
## quality               -0.34835102  0.109362077  0.03331897  0.44036918
##                           quality
## fixed_acidity         -0.08448545
## volatile_acidity      -0.19656168
## citric_acid            0.01833273
## residual_sugar        -0.08206979
## chlorides             -0.31448848
## free_sulfur_dioxide    0.02371338
## total_sulfur_dioxide  -0.19668029
## density               -0.34835102
## pH                     0.10936208
## sulphates              0.03331897
## alcohol                0.44036918
## quality                1.00000000
```

LA columna más interesante es la de la correlación entre la calidad del vino y el resto de variables, donde podemos ver que los valores de correlación son moderados y que se corresponden con las suposiciones que

habíamos realizado en los boxplots del análisis descriptivo del apardato 4.1.

4.6 Modelos de regresión lineal.

En base a los resultados obtenidos en las comparaciones anteriores poremos ver que las diferencias entre los vinos tintos y blancos son importantes, lo cual ya era esperable al ver la elevada cantidad de outliers observados al tratar todos los datos en conjunto.

Por tanto, la mejor estrategia será crear un modelo de regresión para los vinos blancos y otro distinto para los vinos tintos.

En estos modelos de regresión se estimará como variable dependiente el valor de la calidad del vino "quality", como una combinación lineal del resto de variables.

```
#Vinos blancos.
m_white = lm(quality ~ fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + fr

summary(m_white)
```

```
##
## Call:
## lm(formula = quality ~ fixed_acidity + volatile_acidity + citric_acid +
##     residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol, data = data_white)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8348 -0.4934 -0.0379  0.4637  3.1143
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.877909   0.010736 547.502  < 2e-16 ***
## fixed_acidity         0.055290   0.017615   3.139  0.00171 **
## volatile_acidity     -0.187798   0.011470 -16.373  < 2e-16 ***
## citric_acid           0.002673   0.011590   0.231  0.81759
## residual_sugar        0.413285   0.038179  10.825  < 2e-16 ***
## chlorides            -0.005402   0.011941  -0.452  0.65097
## free_sulfur_dioxide   0.063484   0.014357   4.422 9.99e-06 ***
## total_sulfur_dioxide -0.012144   0.016067  -0.756  0.44979
## density              -0.449486   0.057050  -7.879 4.04e-15 ***
## pH                    0.103638   0.015912   6.513 8.10e-11 ***
## sulphates             0.072068   0.011457   6.291 3.44e-10 ***
## alcohol               0.238095   0.029807   7.988 1.70e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7514 on 4886 degrees of freedom
## Multiple R-squared:  0.2819, Adjusted R-squared:  0.2803
## F-statistic: 174.3 on 11 and 4886 DF,  p-value: < 2.2e-16
```

```
#Vinos tintos.
m_red = lm(quality ~ fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + free

summary(m_red)
```

```
##
## Call:
```

```
## lm(formula = quality ~ fixed_acidity + volatile_acidity + citric_acid +
##     residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol, data = data_red)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68911 -0.36652 -0.04699  0.45202  2.02498
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.63602    0.01621 347.788  < 2e-16 ***
## fixed_acidity          0.04351    0.04518   0.963   0.3357
## volatile_acidity      -0.19403    0.02168  -8.948  < 2e-16 ***
## citric_acid           -0.03556    0.02867  -1.240   0.2150
## residual_sugar         0.02303    0.02115   1.089   0.2765
## chlorides             -0.08821    0.01973  -4.470 8.37e-06 ***
## free_sulfur_dioxide    0.04562    0.02271   2.009   0.0447 *
## total_sulfur_dioxide -0.10739    0.02397  -4.480 8.00e-06 ***
## density               -0.03375    0.04083  -0.827   0.4086
## pH                    -0.06386    0.02958  -2.159   0.0310 *
## sulphates              0.15533    0.01938   8.014 2.13e-15 ***
## alcohol                0.29433    0.02822  10.429  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.648 on 1587 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

Como podemos ver R^2 no es muy elevado en ninguno de los tipos de vino, el modelo por tanto no es demasiado bueno. A pesar de ello, el p-valor de la regresión es $<< 0.05$, lo cual nos indica que estadísticamente podemos asegurar que existe algún tipo de relación lineal entre la calidad del vino y el resto de parámetros.

4.7 Modelos de regresión logística propuestos.

Dado que hemos añadido la variable dicotómica class al dataset, podemos llevar a cabo una regresión logística y emplear este modelo a modo de clasificador.

```
#Vinos blancos.
m_log_white = glm(class ~ fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides +

summary(m_log_white)
```

```
##
## Call:
## glm(formula = class ~ fixed_acidity + volatile_acidity + citric_acid +
##     residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol, family = "binomial",
##     data = data[which(data$style == "white"), ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1436  -0.6725  -0.4114  -0.1798   2.8331
##
## Coefficients:
```

```
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.42469    0.11092 -21.859  < 2e-16 ***
## fixed_acidity         0.71577    0.11737   6.099 1.07e-09 ***
## volatile_acidity     -0.62313    0.08042  -7.749 9.28e-15 ***
## citric_acid          -0.10721    0.05827  -1.840 0.065776 .
## residual_sugar        1.40448    0.16955   8.283  < 2e-16 ***
## chlorides            -0.44282    0.13370  -3.312 0.000926 ***
## free_sulfur_dioxide   0.15344    0.05556   2.762 0.005749 **
## total_sulfur_dioxide -0.01524    0.08513  -0.179 0.857936
## density              -1.97635    0.28606  -6.909 4.89e-12 ***
## pH                    0.53751    0.06863   7.832 4.81e-15 ***
## sulphates             0.32258    0.05171   6.238 4.42e-10 ***
## alcohol               0.16978    0.13583   1.250 0.211334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5116.8  on 4897  degrees of freedom
## Residual deviance: 4143.2  on 4886  degrees of freedom
## AIC: 4167.2
##
## Number of Fisher Scoring iterations: 6
```

```r
#Vinos tintos.
m_log_red = glm(class ~ fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + f

summary(m_log_red)
```

```
##
## Call:
## glm(formula = class ~ fixed_acidity + volatile_acidity + citric_acid +
##     residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol, family = "binomial",
##     data = data[which(data$style == "red"), ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9878  -0.4351  -0.2207  -0.1222   2.9869
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.53572    0.38803  -6.535 6.37e-11 ***
## fixed_acidity         0.35646    0.16242   2.195 0.028183 *
## volatile_acidity     -0.42493    0.12912  -3.291 0.000999 ***
## citric_acid           0.08251    0.12185   0.677 0.498313
## residual_sugar        1.13932    0.35081   3.248 0.001163 **
## chlorides            -0.30887    0.11788  -2.620 0.008788 **
## free_sulfur_dioxide   0.19206    0.21716   0.884 0.376469
## total_sulfur_dioxide -0.93434    0.27661  -3.378 0.000731 ***
## density              -0.77305    0.33105  -2.335 0.019536 *
## pH                    0.03605    0.16052   0.225 0.822327
## sulphates             0.55800    0.08059   6.924 4.39e-12 ***
## alcohol               0.89852    0.15697   5.724 1.04e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1269.92  on 1598  degrees of freedom
## Residual deviance:  870.86  on 1587  degrees of freedom
## AIC: 894.86
##
## Number of Fisher Scoring iterations: 6
```

Como podemos ver para el modelo white el valor de AIC es enorme, unos 4000, para el modelo red es algo menor sobre los 900.

Vamos a calcular la exactitud del modelo clasificador:

```
#Cálculo exactitud vinos tintos.
predict_red<- predict.glm(m_log_red, data[which(data$style=="red"),-c(12,13)], type="response")
predict_red[predict_red >= 0.5]<-"good"
predict_red[predict_red < 0.5]<-"bad"

table(data$class[which(data$style=="red")],predict_red)
```

```
##       predict_red
##         bad good
##   bad  1339   43
##   good  142   75
```

La exactitud es de un 88% para los vinos tintos, no es un mal resultado. Donde más falla el modelo es en la clasificación de los buenos vinos (quality >= 7) ya que solamente acierta en un 34%.

```
#Cálculo exactitud vinos blancos.
predict_white<- predict.glm(m_log_white, data[which(data$style=="white"),-c(12,13)], type="response")
predict_white[predict_white >= 0.5]<-"good"
predict_white[predict_white < 0.5]<-"bad"

table(data$class[which(data$style=="white")],predict_white)
```

```
##       predict_white
##        2.07168416895837e-07  bad good
##   bad                     1 3632  205
##   good                    0  763  297
```

La exactitud es de un 80% para los vinos blancos, tampoco es un mal resultado. Donde más falla el modelo es en la clasificación de los buenos vinos ya que solamente acierta en un 28%.

4.8 Modelos supervisados propuestos (Random Forest).

Generaremos dos modelos de clasificación supervisados de tipo random forest, para vinos blancos y tintos.

Dividiremos el dataset en un train y test sets en una proporción de 2/3, 1/3. Para generar el modelo con el train set, realizaremos una validación cruzada de 4 folds.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(rminer)
```

```r
#Separamos del dataframe data (contiene class y style), por tipo de vinos, eliminamos el parámetro qual
datar<-data[which(data$style=="red"), c(-12,-13)]
dataw<-data[which(data$style=="white"), c(-12,-13)]

#Clasificación random forest vinos tintos.
h <- holdout(datar$class,ratio=2/3, mode="stratified")
data_train <- data[h$tr,]
data_test <- data[h$ts,]
#Para entrenar el modelo realizamos una validación cruzada de 4 folds.
train_control<- trainControl(method="cv", number=4)
mod<-train(class~., data=data_train, method="rf", trControl = train_control)
pred <- predict(mod, newdata=data_test)

confusionMatrix(pred,data_test$class,positive="good")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad good
##       bad  461    0
##       good   0   72
##
##                Accuracy : 1
##                  95% CI : (0.9931, 1)
##     No Information Rate : 0.8649
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.1351
##          Detection Rate : 0.1351
##    Detection Prevalence : 0.1351
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : good
##
```

```r
#Clasificación random forest vinos blancos.
h1 <- holdout(dataw$class,ratio=2/3, mode="stratified")
data_train1 <- data[h1$tr,]
data_test1 <- data[h1$ts,]
#Para entrenar el modelo realizamos una validación cruzada de 4 folds.
train_control1<- trainControl(method="cv", number=4)
mod1<-train(class~., data=data_train1, method="rf", trControl = train_control1)
pred1 <- predict(mod1, newdata=data_test1)

confusionMatrix(pred1,data_test1$class,positive="good")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  bad good
##       bad  1333    0
##       good    0  299
##
##                Accuracy : 1
##                  95% CI : (0.9977, 1)
##     No Information Rate : 0.8168
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.1832
##          Detection Rate : 0.1832
##    Detection Prevalence : 0.1832
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : good
##
```

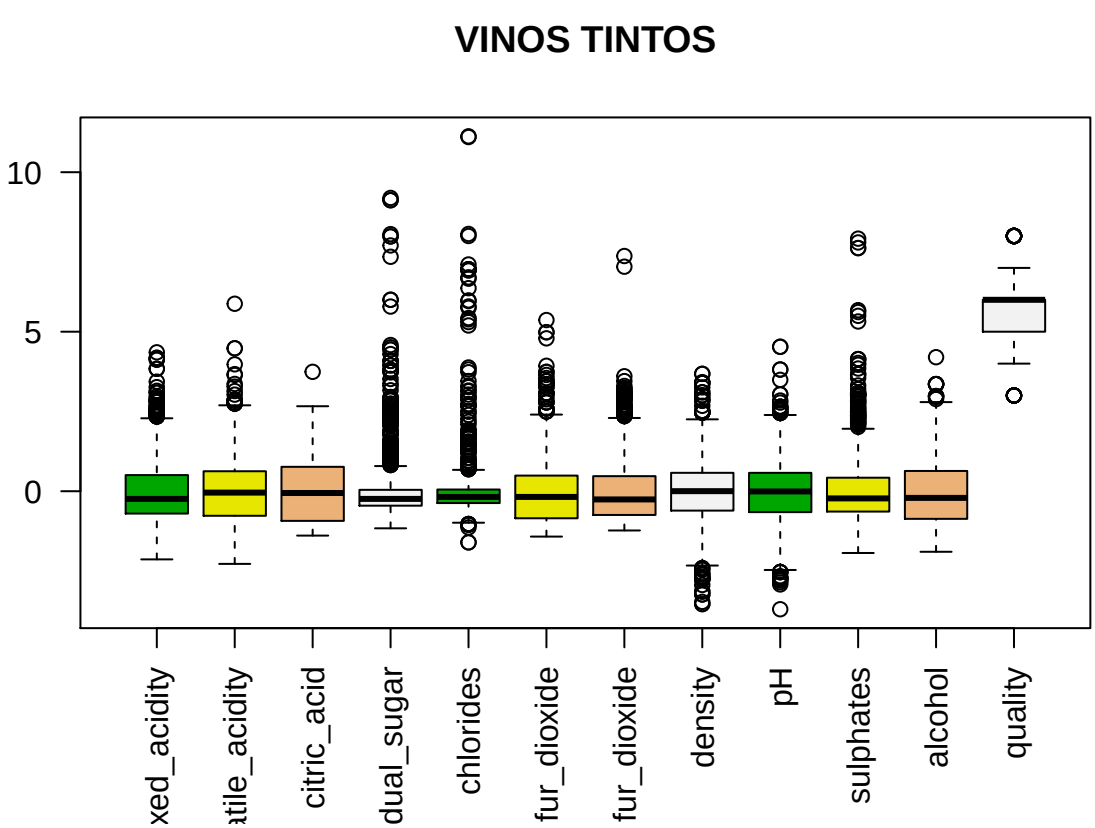
Los resultados de la clasificación son demasiado buenos, en ambos casos clasifica perfectamente ambos tipos de vinos. Claramente he cometido algún error en el modelo, ya que modelos que clasifiquen con un 100% de exactitud son muy improbables, he revisado el procedimiento en repatidas ocasiones pero no he sido capaz de detectar el fallo.

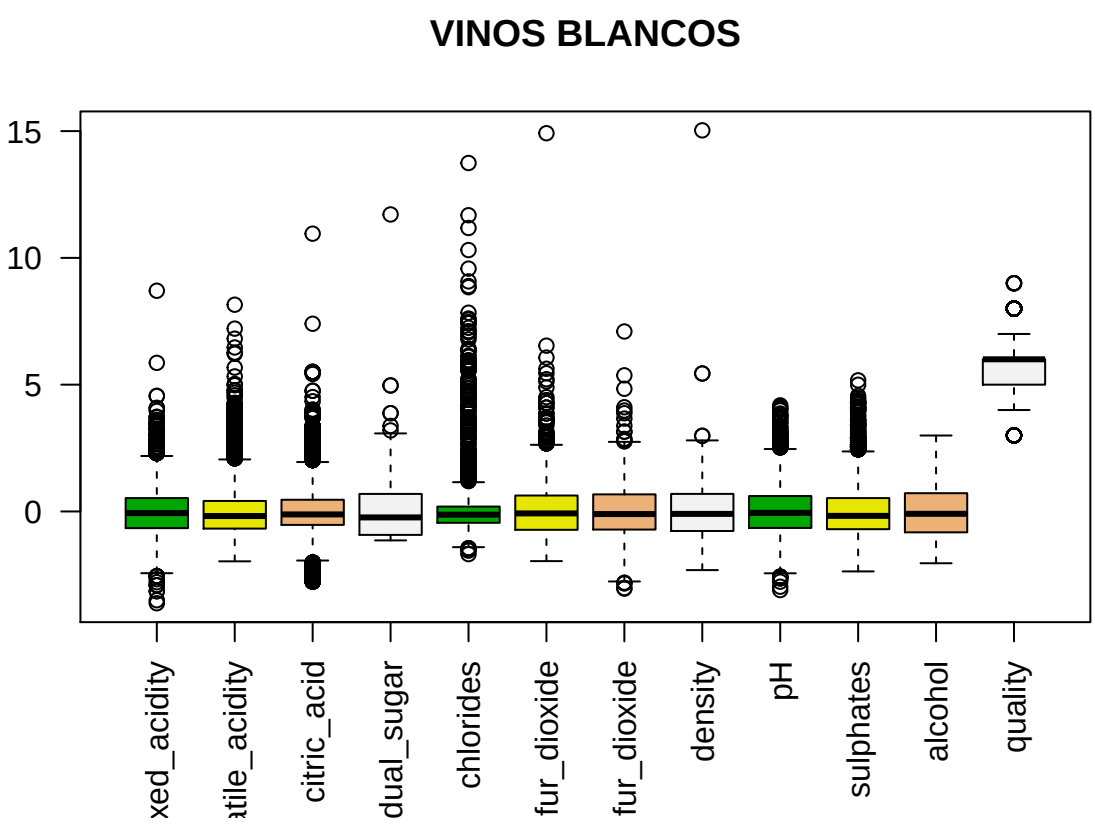
5 Representación de los resultados.

Procedemos a representar los resultados obtenidos en los análisis.

5.1 Boxplots de los atributos de los vinos blancos y tintos.

También forman parte de esta representación los boxplots generados en el apartado 4.1 donde se han generado comparativos entre buenos y malos para cada atributo y cada tipo de vino (tinto o blanco), no los vuelvo a insertar para no alargar en exceso el apartado.


VINOS TINTOS
10
5
0
xed_acidity
tile_acidity
citric_acid
dual_sugar
chlorides
fur_dioxide
fur_dioxide
density
pH
sulphates
alcohol
quality



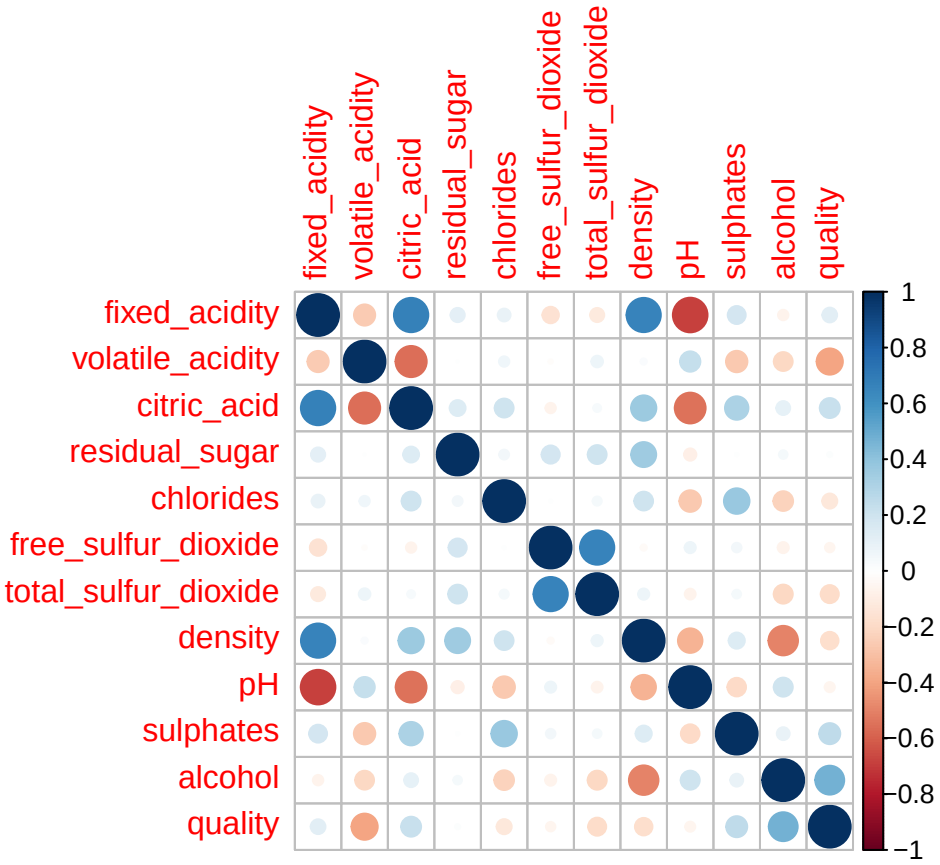
5.2 Correlaciones.

Correlación entre variables vinos tintos y blancos, donde podemos ver que las variables no están muy correlacionadas entre si.

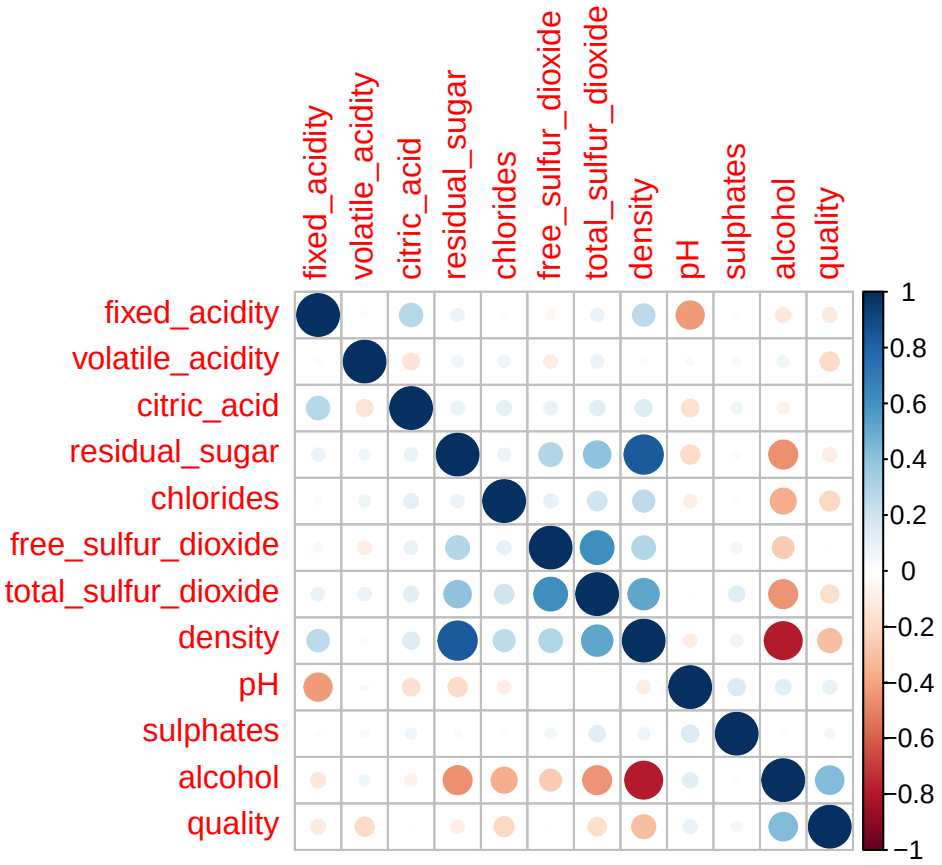
```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
#VINOS TINTOS.
corr.res<-cor(data_red[,-c(13,14)])
corrplot(corr.res,method="circle")
```




```
#VINOS BLANCOS.
corr.res1<-cor(data_white[,-c(13,14)])
corrplot(corr.res1,method="circle")
```



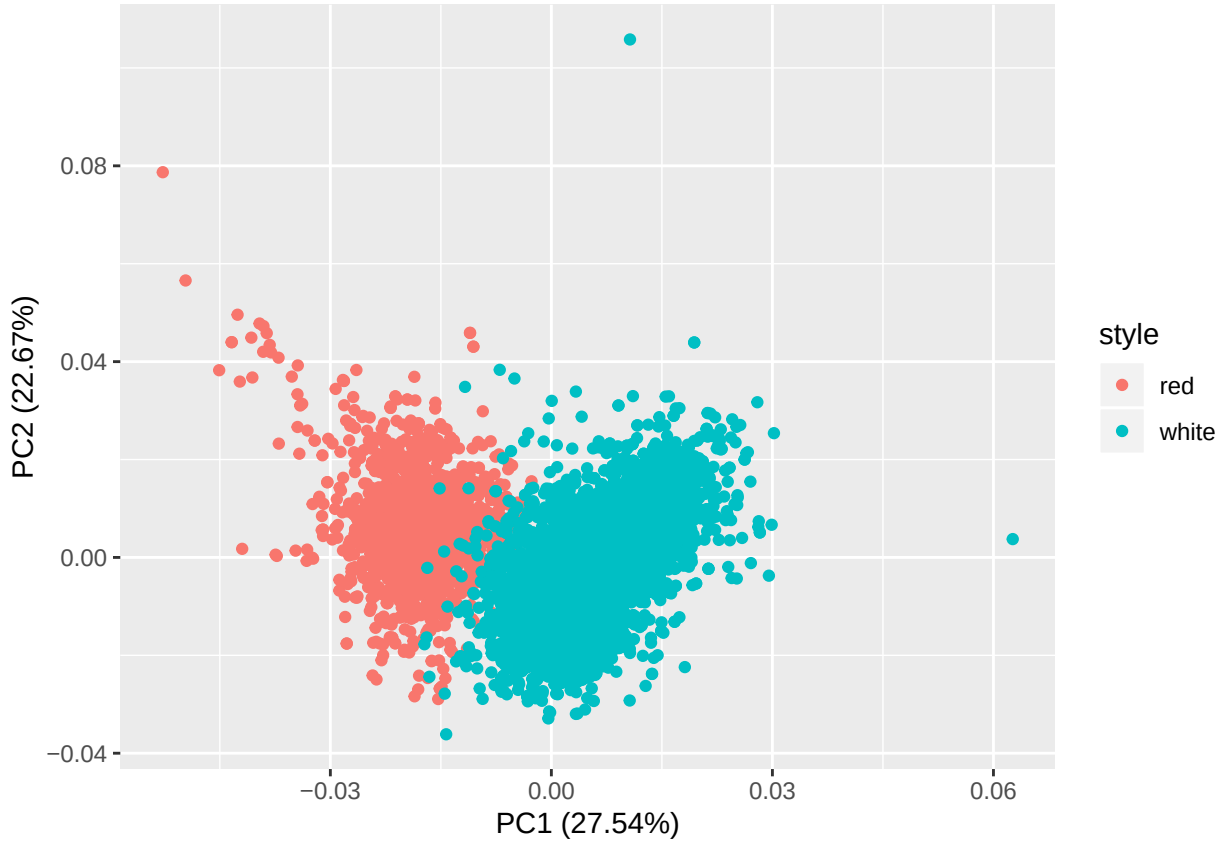
5.3 Tabla resumen tests estadísticos.

Tabla resumen con los resultados estadísticos, donde podemos ver que estadísticamente entre los atributos de tintos y blancos y vinos buenos y malos, existen diferencias significativas entre la mayoría de ellos, aunque ya hemos apuntado que la cantidad elevada de outliers reduce bastante la potencia estadística de los resultados.

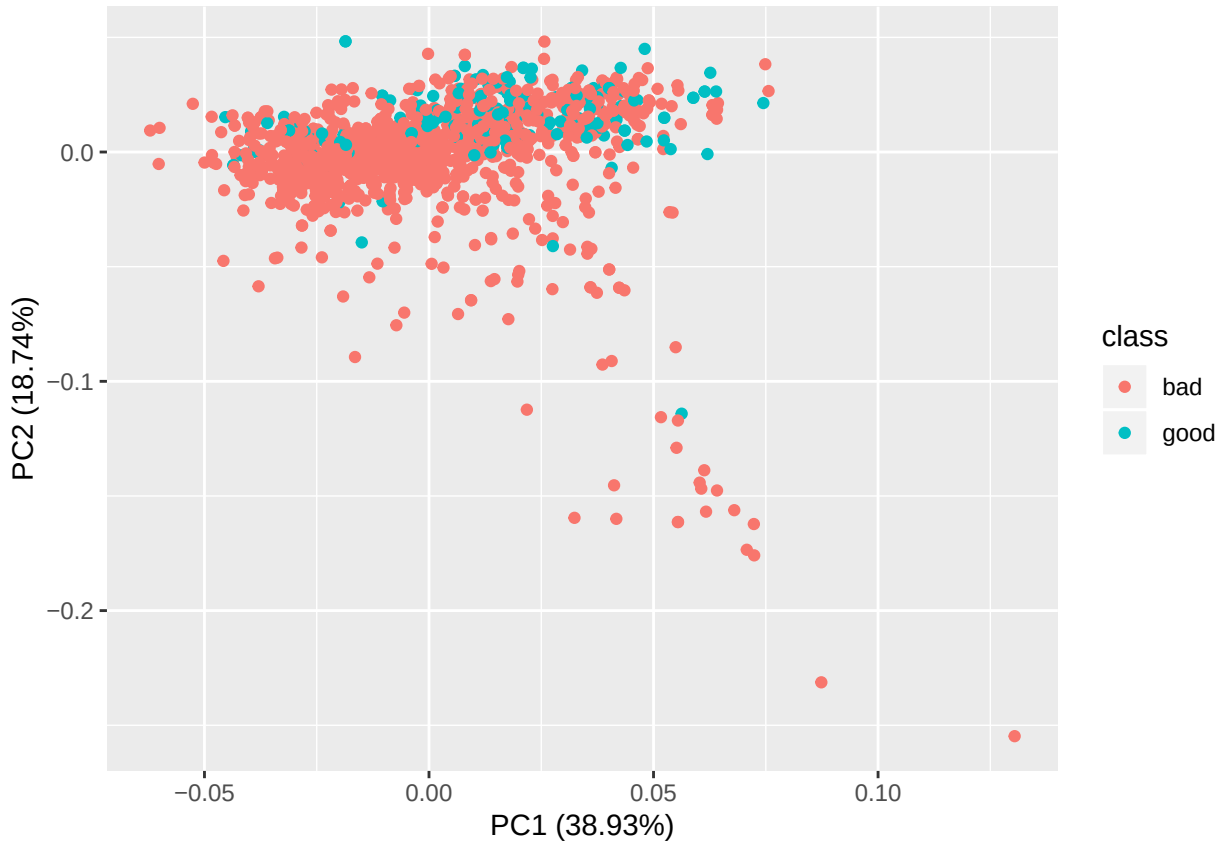
Variable	Diferencias significativas entre tintos y blancos (style Red-White)	Diferencias significativas vinos tintos entre buenos y malos (class Good-Bad)	Diferencias significativas vinos blancos entre buenos y malos (class Good-Bad)
fixed_acidity	SI	SI	SI
volatile_acidity	SI	SI	SI
citric_acid	SI	SI	NO
residual_sugar	SI	SI	SI
chlorides	SI	SI	SI
free_SO2	SI	SI	NO
total_SO2	SI	SI	SI
density	SI	SI	SI
pH	SI	SI	SI
sulphates	SI	SI	NO
alcohol	NO	SI	SI
quality	SI		

5.4 Proyecciones PCA:

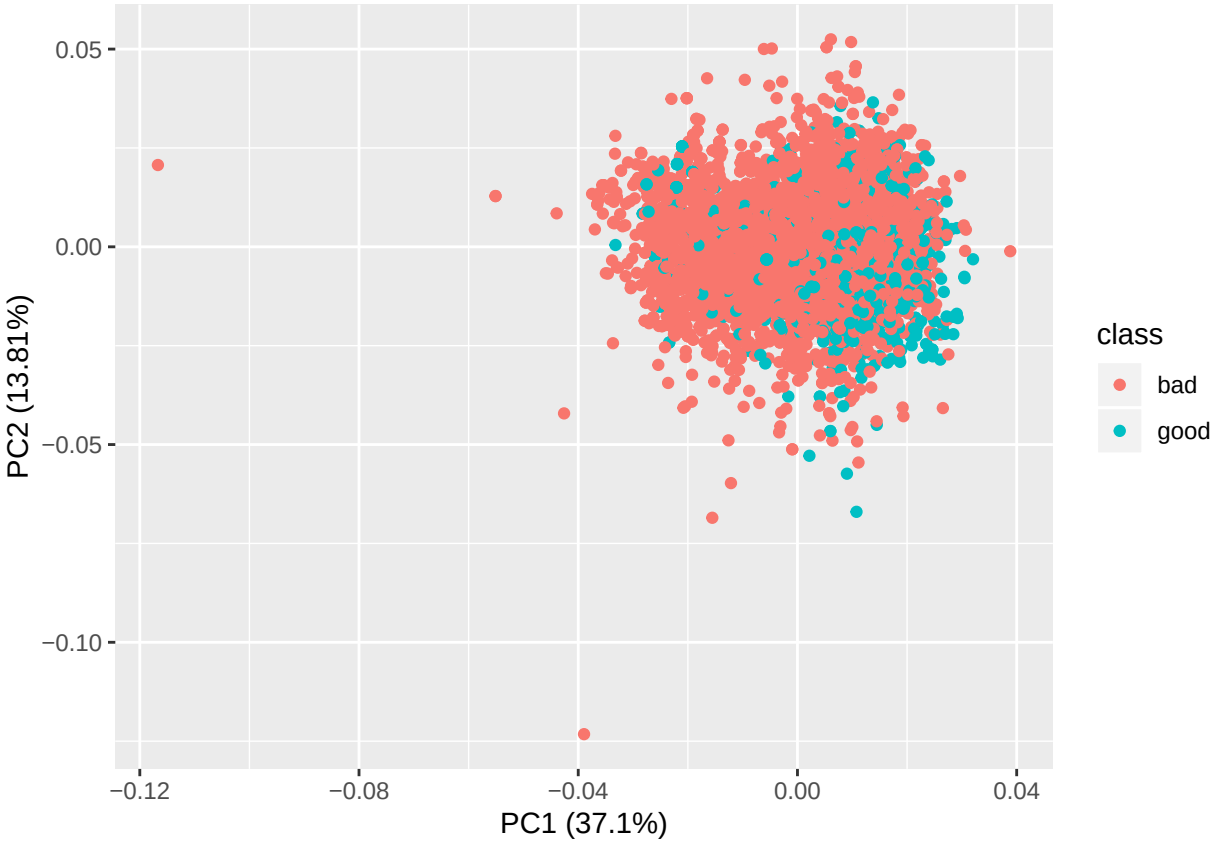
En este primer gráfico representamos los dos primeros PC (PC1 y PC2), los cuales acumulan un total del 50% del la varianza de los datos, y a pesar de que este % de varianza no es mucho, ya podemos ver como los vinos blancos se separan claramente de los tintos.



En esta segundo gráfico, representamos la proyección de las dos primeras componentes principales PC1 y PC2, pero esta vez del dataset de vinos tintos, para ver si los vinos de buena calidad se separan de los de mala calidad. Como podemos ver, con un 55% de la varianza de los datos acumulada, parece que los vinos de buena calidad se desplazan a una región del gráfico, aunque no se separan con claridad de los vinos de mala calidad.



Con los vinos blancos sucede algo parecido, aunque parece que se separan mejor (ambas PC acumulan aprox. un 47% de la varianza de los datos).



5.5 Tablas resumen modelos de análisis.

De los análisis realizados, los resultados son los siguientes:

MODELO REGRESIÓN LINEAL

Modelo lineal que correlaciona el nivel de calidad con el resto de atributos del vino:

VARIEDAD	R^2	p-value	Atributos con mayor incidencia(coeficiente)	Atributos sin incidencia estadística en el modelo Pr(>/t/)>0.05
BLANCOS	0.28	<2.2e-16 Estadística-mente se puede afirmar qu existe una correlación lineal	volatile_acidity(-0.19), residual_sugar(0.42), density(-0.44), pH(0.10), alcohol(0.23)	citric_acid(0.81), chlorides(0.65), total_SO2(0.45)

VARIEDAD	R^2	p-value	Atributos con mayor incidencia(coeficiente)	Atributos sin incidencia estadística en el modelo Pr(>/t/)>0.05
TINTOS	0.36	<2.2e-16 Estadísticamente se puede afirmar qu existe una correlación lineal	volatile_acidity(-0.19), total_SO2(-0.11), sulphates(0.16) alcohol(0.29)	fixed_acidity(0.33), citric_acid(0.22),residual_sugar(0.28), density(0.41)

MODELOS CLASIFICADORES

Si tomamos como clase de interes los vinos de buena calidad (calidad >=7).

VINOS TINTOS

MODELO	Exactitud	Sensibilidad	Especificidad
Regresión logística	88%	28%	97%
Random Forest	100%	100%	100%

VINOS BLANCOS

MODELO	Exactitud	Sensibilidad	Especificidad
Regresión logística	88%	35%	95%
Random Forest	100%	100%	100%

Los valores de calidad del modelo de random forest son demasiado elevados, casi con toda seguridad he cometido algún error en el modelado de los datos mediante rf.

5.6 Conclusiones, resolución a las preguntas planteadas.

Una vez evaluados resultados de los análisis realizados, pasamos a contestar a las preguntas planteadas inicialmente:

- ** ¿Tiene sentido trabajar con todos los vinos en conjunto o es mejor trabajar con ellos de forma separada (blancos respecto a tintos)?,¿son significativamente distintos los vinos blancos de los tintos?**

Tal como hemos podido comprobar ambos tipos de vinos son distintos de forma significativa estadísticamente, por lo que es mejor trabajar con los dos datasets (blancos y tintos) por separado.

- ** ¿Existe algún parámetro químico físico que sea especialmente relevante para diferenciar vinos de buena calidad?**

En los modelos de regresión podemos ver que existe estadísticamente una clara correlación lineal (p-valor<2.2e-16), aunque el valor de R^2 es bastante bajo. En estos modelos lineales, así como en las matrices de correlación, aparecen algunos atributos que parecen tener una mayor incidencia en el modelo lineal, como son el alcohol, volatile_acidity y alguno más (ver apartado 5.5) aunque sus coeficientes tampoco tienen un peso muy elevado.

- ** ¿Es posible predecir, por ejemplo mediante regresión lineal, u otros tipos de análisis, en base a los datos químico físicos disponibles de un vino, determinar la calidad resultante del mismo antes de que lo valore un panel de expertos?, ¿Cual sería el mejor de estos métodos de análisis?**

Es posible hacer predicciones de calidad con el modelo lineal, y por ejemplo con el modelo de regresión logística o con el modelo supervisado random forest (como ya he comentado los resultados con este modelo son bastante dudosos y seguramente son erróneos).

A pesar de ello, se puede observar (tanto en los modelos de regresión lineal como logística), que su sensibilidad es bastante baja, o sea que los vinos de calidad regular son clasificados de manera correcta, pero en los vinos de buena calidad los modelos cometen muchos errores y en un % elevado terminan clasificando buenos vinos como vinos mediocres. Por tanto a pesar de tener una buena exactitud, el defecto de estos modelos es su baja sensibilidad.

Conclusiones adicionales, posibles mejoras.

En base a estas conclusiones, se podrían proponer posibles mejoras al dataset con el objeto de elevar los parámetros de calidad de los diferentes modelos, por ejemplo algunas de ellas podrían ser:

- Incrementar el número de vinos de buena calidad (quality >=7) en el dataset, su número es muy reducido y esto puede ser una de las causas de la baja sensibilidad detectada en los modelos.
- La cantidad de outliers existentes en los diferentes atributos del dataset es elevada, lo cual reduce la potencia estadística de los resultados. Ya he apuntado que una de las posibles causas es que la producción de un vino es un proceso semiartesanal muy dependiente de las condiciones climatológicas de un año determinado y también de la región donde este vino se produce (características del suelo y del microclima de la región), por lo que podría ser una buena idea añadir al dataset original, datos de localización geográfica de cada uno de los vinos, y de las condiciones climatológicas del año en que se relizó la cosecha.